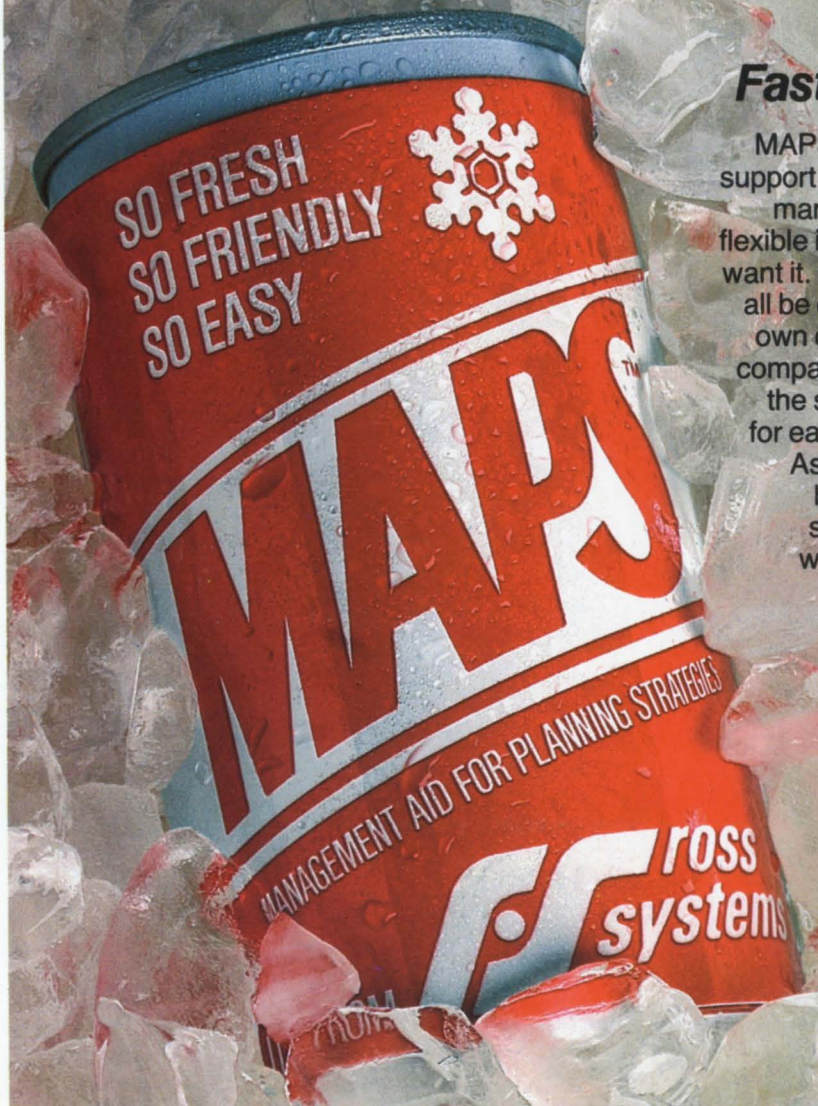
RSTS PROFESSIONAL

Volume 4, Number 3

June 1982
$10.00/issue, $35.00/year

INSIDE:

Financial Decision Support Canned For Easy Use!



Fast Flexible Information

MAPS™ IS FRESH! It's the ideal decision support software product for today's financial manager. With MAPS you get fast, fresh, flexible information handling the moment you want it. Plans, forecasts, models, reports can all be created and manipulated right in your own department. Changes that affect your company can be reflected and evaluated on the spot. And best of all, MAPS is canned for easy use! Call Ross Systems right now. Ask for a demonstration of MAPS. You'll be amazed what a little fresh decision support can do. Currently available on worldwide timesharing, or purchase for DEC's,* RSTS/E and VMS operating systems. Contact Ross Systems for more information today.

*DEC is a trademark of Digital Equipment Corporation.

ross systems

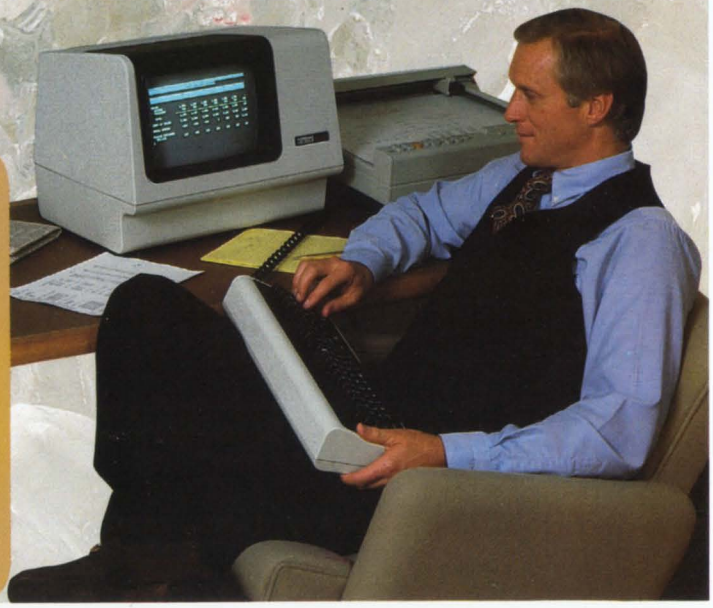
1800 Embarcadero Road
Palo Alto, CA 94303
(415) 856-1100

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

"Now that's fresh decision support."

MAPS meets your daily requirements:

Financial reports	100%
Financial planning	100%
Consolidations	100%
Financial modeling	100%
Performance forecasts	100%
Full color graphics	100%

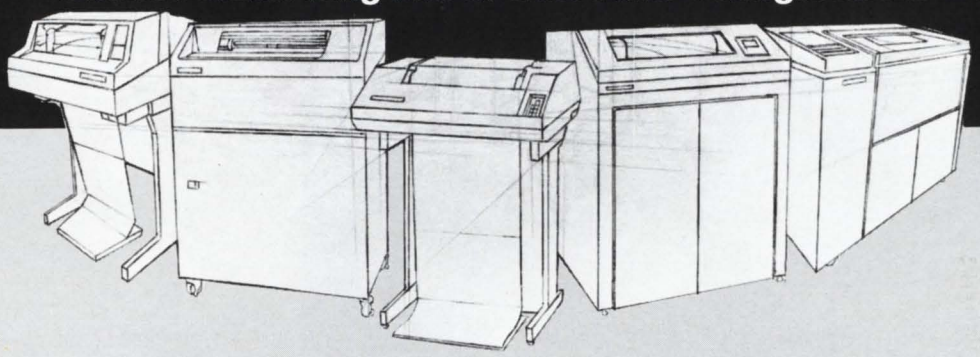


CIRCLE 1 ON READER CARD

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

Compatibility guaranteed with all DEC processors!

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



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

Southern Systems

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

My computer system is a

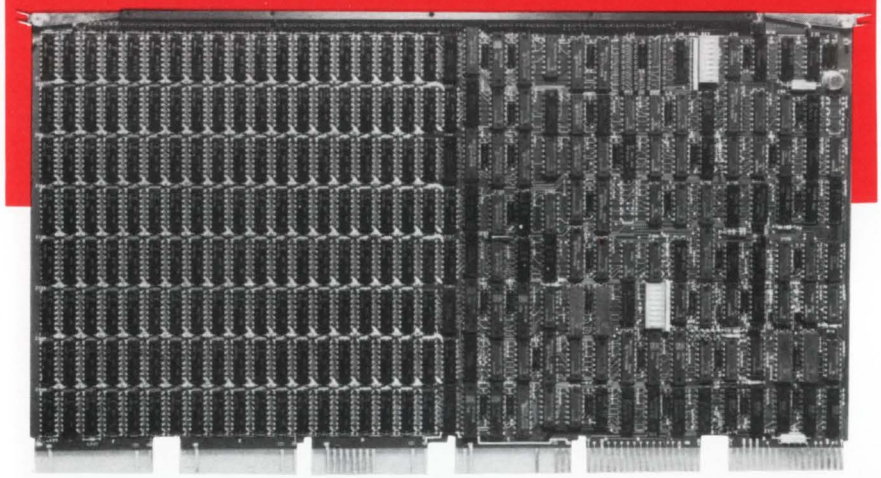
_____

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

Name_____
Title_____
Company_____
Address_____
City_____
State_____ Zip_____
Telephone_____

CIRCLE 2 ON READER CARD

DEC®-COMPATIBLE SEMI ADD-INS



See Us At
COMPEX
JUNE 28-30, 1982
Atlantic City Convention Hall
Atlantic City, NJ

FROM THE LEADER

Look to the leader — Dataram — for your DEC-compatible semiconductor add-in memory. Offering not only the broadest, most complete line of semi add-ins, but the most capable...no matter what your yardstick. Compatibility, throughput, cost, power efficiency, size...no matter how you measure capability, Dataram DEC-compatible semi add-ins are the clear leader.

A leadership position earned by improving on DEC's price and delivery...and then adding features available from no one else in the industry.

The chart provides a glimpse at the industry-pacesetting family of DEC-compatible semi add-ins. Circle the reader service number below or, better yet, call us today at 609-799-0071, and we'll give you a close-up look at the products that have made us the leader.

DEC Mini	Dataram Add-In	Board Size	Maximum Capacity
LSI-11®	DR-115S	dual	64 KB
LSI-11	DR-215S	dual	256 KB
LSI-11	DR-113S	quad	256 KB
LSI-11	DR-213S	quad	1.0 MB
PDP®-11	DR-114S	hex	256 KB
PDP-11	DR-114SP	hex	256 KB
PDP-11	DR-214SP	hex	1.0 MB
PDP-11	DR-144S	hex	256 KB
PDP-11	DR-244S	hex	4.0 MB
VAX®-11/750 PDP-11/70	DR-175S	hex	256 KB
VAX-11/780	DR-178S	extended hex	512 KB
DECSYSTEM 2020®	DR-120S	extended hex	512 KB
PDP-8/A	DR-118S	quint	128 K x 12

DEC, DECSYSTEM 2020, PDP and VAX are registered trademarks of Digital Equipment Corporation.

Dataram also provides core add-ins, core and semiconductor add-ons, memory system units, memory management, and a wide range of memory-related accessories for DEC users.

DATARAM CORPORATION

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

CIRCLE 55 ON READER CARD

Contents

Coming . . .

- TYPE.RTS
- Word Processing
- V7.1 and 7.2 Report
- CB
- The DEC Professional
- Crystal Ball Part 2
- VAX/VMS Internals
- Data Communications
- PACMAN on RSTS
- More EDT
- Performance Measurement
- DATATRIEVE - 32
- RSTS Security
- More . . .

The RSTS Professional Magazine, June 1, 1982, Vol. 4, No. 3. Published bi-monthly. Single copy price $10.00, $35.00 per year. Published by M Systems, Inc., 753 Johns Lane, Ambler, Pa. 19002-0245, telephone (215) 542-7008. Send all correspondence and change of address to: Box 361, Ft. Washington, Pa. 19034-0361.

From the editors. . .

Carl B. Marbach

I have just returned from the Spring DECUS meetings in Atlanta and I thought I might share a few random reactions with you. DEXPO-82, the DEC compatible show proved to be a big success and I take my hat off to Larry Hollander of ExpoConsul who conceived the idea, promoted it, sold it out and brought it off as a real professional exposition; great job! I was surprised by the calm with which DECUS accepted this fait accompli, after all, he violated most DECUS rules regarding commercialism. Even the closed circuit TV system in the DECUS hotel advertised that attendees should come over to visit DEXPO! DEXPO isn't a member of DECUS and I guess that gives him some privileges that members don't have. Next for DEXPO will be DEXPO WEST matching the Anaheim meeting later this year. From what I saw it will be bigger than Atlanta. How will we ever find time to give papers, attend papers, attend meetings, talk in the halls AND go to DEXPO?

Questions: Why was the NEW USER paper scheduled for the last evening of the week? Why is it "neat" that some "children" broke into the RSTS demo system and deliberately crashed the monitor? Why do most DECUS people still say "no" and "we can't do it that way". Why does the Hilton hotel hosting the DECUS meeting run out of rooms, even for people with paid reservations? And then why couldn't they let me eat lunch and check out 1 hour late? What ever happened to "real" hotels where you were a "guest" not a body? How can 8 elevators take so long to move so few people?

We want to have a SIG party. UNofficial. Here's how we propose to do it: Dave and I are going to give a paper at Anaheim titled, "Hints and Tips for INTERMEDIATE RSTS/E Users". This will not be a beginner session but not a guru one either. We will try to have it from 8:00 P.M. till ??? and maybe even have some refreshments in the back for after 11:00. We'll try to schedule it — you try to come.

The RSTS PROFESSIONAL will try to give a NEW USER session as well as the one mentioned above, if we can schedule it properly for new users and first time DECUS attendees. If you've never been to a DECUS, try it; it will be the best thing you can do for yourself and your company. Why not consider giving a paper while you're there; it isn't hard and your fellow RSTS users will appreciate it. Most of us have one or two things we can teach the other guy and I never fail to learn something every day of DECUS.

The RSTS PROFESSIONAL is being joined by a new publication, the DEC PROFESSIONAL. This new journal will be in ADDITION to the RSTS PRO and will come out during the months the RSTS PRO doesn't. It will be (we hope) as good and solid a technical journal as we have seen the RSTS PRO become. The DEC PROFESSIONAL will have sections on all of DEC's operating systems and be a more general magazine than this one. It will initially be distributed free to over 50,000 DEC users all over the world. We are looking for authors and if you have something to say to the entire DEC community we will consider publishing it. There is a moderate remuneration for articles accepted for publication and you can become famous besides! Look for the DEC PROFESSIONAL beginning mid-summer of 1982.

SECURITY/INSECURITY

Dave Mallery

We have a big problem in the RSTS community. First in LA and now in Atlanta the same group of RSTS break-in artists masquerading as security "experts" have ruined both the SIG's credibility and possibily the future availability of RSTS machines for DECUS meetings. This mess was achieved by breaking into and ruining the system on several occasions until the folks in charge simply shut it off in disgust.

It seems that there are two major problems, both caused by naivete:

1) The people doing the breaking think that somehow we will admire or fear their prowess and hire them to protect our systems.

2) The people setting up the RSTS system for the demo room did not learn from the LA experience and by repeating the same errors (not all theirs), have left the product's integrity in doubt.

Simply banning these "experts" from future Symposia will not work as well as setting up a correctly secured (by the book) system and watching them beat their brains out against it.

RSTS PROFESSIONAL

Editors
R. D. Mallery
Carl B. Marbach
Assistant Editor/Advertising
Helen B. Marbach
Assistant Editor/Operations
Peg Leiby
Administrative Assistant
Hope Makransky
Subscription Fulfillment
Kathi B. Campione
Claire Hollister
United Kingdom Representative
Pauline Noakes
RTZ Computer Services Ltd.
P.O. Box 19, 1 Redcliff Street
Bristol, BS99-7JS
Phone: Bristol 24181
Contributors
Tom Britton
Mark J. Diaz
Eric Dickman
R. A. Dudley
Wef Fleischman
Andrew G. Gault
Michael C. Greenspon
Patrick Holmay
Philip Hunt
Michael H. Koplitz
Richard Davis Mallery
Bob Meyer
W. Franklin Mitchell, Jr.
Dennis Morgan
Paul O'Nolan
R. A. Smith
David Spencer
Jude Suszko
Cartoons
Douglas Benoit
Game Verification Team
Bill Marbach, Mary Marbach,
Sydra Mallery, Sam Mallery
Design & Production
Grossman Graphics
Printing & Binding
Schneider Litho Co., Inc.

LOW COST RSTS/E NETWORKING.

DMG/NET provides RSTS/E users with easy access to packet-switched (X.25) networks. It permits two-way file transfer and interactive dialogue with other RSTS/E systems and locally initiated communication with non-RSTS/E systems. From a RSTS/E host to other RSTS/E computers, to other DEC computers, even to non-DEC computers...communication is quick, simple and extremely inexpensive.

REDUCES COSTS

No matter how large or how small your RSTS/E system is, DMG/NET can substantially reduce communication costs by utilizing packet-switched (X.25) networks. With DMG/NET, you can save up to 90% of your cost of communication, compared to "dial-up" or leased lines.

EASY TO USE

Even a non-technical person can access the entire communication network quickly and easily with DMG/NET. There is no need to memorize a long series of network codes, numbers and procedures. Instead, a user accesses a remote database with only the familiar local sign-on and a short, easily remembered identification code...DMG/NET does the rest.

What's more, DMG/NET allows any locally connected terminal to access any specified remote database, eliminating the need for separate terminals, complicated switch boxes or terminal setting changes.

To find out how DMG/NET can meet *your* RSTS/E networking needs, contact Digital Management Group Ltd.

DMG/NET

Digital Management Group Ltd.
4800 Yonge Street, Willowdale, Ontario, Canada M2N 6G5
Telephone: (416) 225-7788

CIRCLE 88 ON READER CARD

LETTERS to the RSTS Pro...

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

The enclosed is a note about a performance problem I had recently with one of the RSTS systems I have in my care. The problem was self-inflicted, but all too easily done. Others may find the description and explanation illuminating.

The RSTS Pro is a superb publication! There are always many gems of information liberally sprinkled through the letters, articles, etc. The advertising too is a mine of information. All this information is really great, but how about more rumour?

Tom Britton, Research & Development Mgr.
CBL Canterbury Ltd.
Christchurch, New Zealand

RSTS RUMOURS ... PROFESSIONAL RUMORS ... PROFESSIONAL RSTS RUMOURS! ... sounds like a possibility, Tom, stay tuned.

[*See Mr. Britton's article, "A Shortage of Small Buffers", in this issue.*]

—

With reference to your article entitled "Is It Really Fair?" in the April, 1982 issue of the *"RSTS Professional"*, I would like to express a most hearty AMEN. Further, I would like to state unequivocally that it is not fair what Digital Equipment Corporation (DEC) is doing to it's loyal and extremely large RSTS community. I have been on DEC's back for well in excess of five years concerning its illogical and ineffective approach to the 32-bit world versus the already successful and proven 16-bit approach. Before I continue with my comments, let me take a moment to pass on to you some information about my organization. This should give you an idea as to the impact that DEC's unsolicited approach to the 32-bit world is having relative to health care institutions. The following are statistics relating to the Baptist Memorial Hospital in Memphis, TN.

1. BMH is the largest not-for-profit health care institution in the United States. We are licensed at 2,068 beds. In addition, we presently have acquired or leased six other not-for-profit health care institutions which total in excess of 500 beds.

2. BMH and it's affiliates employ over 6,000 people.

3. Our data processing configuration is one of the purest distributed data processing shops in the country. Yearly we have visitors from all over the world that come to BMH to study our philosophy of distributed data processing.

4. The distributed data processing operation is comprised of the following:

a) 8 Megabyte to megabyte and a half PDP 11/70's running RSTS/E and Basic Plus *only*.

b) 7 One-half to 1 megabyte PDP 11/44's running RSTS/E and Basic Plus *only*.

c) 2 128 KW PDP 11/60's running RSTS/E and Basic Plus *only*.

d) 1 128 KW PDP 11/34 running MUMPS.

e) 1 192 KW DEC 20/20 running TOPS 20.

f) 40 DEC stand alone word processors.

g) 9 of the above PDP 11's in items a), b), and c) are DECNET'ed together.

h) We have a software investment in excess of 4,000 programs, all of which are in *Basic Plus*.

i) The 4,000 programs comprise approximately 79 software applications.

j) Over 450 terminals in operation. Will be in excess of 600 in a year.

NOTE: Items d) and e) are not a part of the hospital's information system, but rather are dedicated for scientific purposes and operations statistical research respectively.

As you can ascertain from the above statistics, we at BMH have a multi-megabuck investment in RSTS/E, Basic Plus, and the PDP 11. It would be a gross understatement to say that we are furious with DEC's obliviousness toward the RSTS community with regards to VAX and in particular the RSX emulator called VMS.

I was fortunate enough to be given the opportunity to view the VAX before it had a name and while it was still in developmental stages. At that time, I provided DEC management with what I felt was the criteria in order to make the VAX a successful commercial product. My statements revolved, in particular, around RSTS/E, Basic Plus, and peripheral enhancements. At that juncture I was informed that VAX/VMS was not intended, at this time, for the commercial world. End of discussion.

Today, as you see, and have pointed out in your article we are being asked to convert to a machine that is unacceptable commercially, costs more, gives you less, and is completely incompatible with the PDP 11, RSTS, Basic Plus environment. Not only that, DEC is discontinuing the 11/70, which, in my opinion is it s jewel, and has no plans that we are aware of to replace it. As I see it, the VAX is still a scientific FORTRAN machine that is being forced upon the RSTS community.

Health care in this country is under fire because of increasing costs. Here at BMH we have one of the lowest per patient day costs in the country primarily because of the PDP11, RSTS/E, and Basic Plus approach. When DEC representatives are confronted with many statistics and facts concerning productivity, transportability, capability and costs relative to the success of the RSTS approach versus the VMS approach, all they can do is seem to make the statement illogically that "VAX/VMS is where the Corporation has decided to put it s money". This is, of course, ignoring the facts and asking us basically to trust them. I do not understand how they can do this without any kind of effective marketing research being accomplished on the front-end and even now. If effective marketing research were accomplished and if DEC would listen to it, they would most certainly and expeditiously abandon VMS for the commercial user and embark upon a project to implement RSTS/E and Basic Plus, completely transportable up and down the 11/family, on the VAX.

If you sense cynicism, bitterness, and sarcasm, my answer to that would be, as Justice Warren so ably said (modified), yes, yes, yes, and it is not fair. As you also might be able to tell, we are just another typical multi-million dollar satisfied DEC customer. It's difficult to be excited about DEC's new announcements when one has just had his legs amputated, corporately speaking.

If you have any thoughts on how we might achieve a greater impact on DEC *upper management* with regards to the RSTS versus VMS controversy, please do not hesitate to let me know how we might gather our forces. My philosophy is one that says *never give up*. If DEC continues its approach down an unlit corridor, I guess that I will be spouting this philosophy even as I am forced to return to one of the major main-frame vendors.

I enjoy the *RSTS Professional* very much. Keep up the good work!

Sincerely, Ron L. Scoggins
Director of Data Processing
Baptist Memorial Hospital, Memphis, TN

!!!!!

—

Congratulations on an excellent publication. I tremble for the consequences if any of your issues should go astray.

Yours sincerely, Geoff Fooks
Bromley, Kent, UK

—

I thought you might be interested in seeing WUGMAG, the newsletter of the UK WORD-11 User Group, especially since we are promoting *"RSTS Pro"* in the current issue. As you can see we have an active User Group here, WUGMAG goes to seventy sites in UK, one in Australia and three in the U.S. even!

At our site we've found *"RSTS Pro"* so useful that I took the opportunity of a spare WUGMAG back page to blow the trumpet, with Pauline's agreement of course. [*Pauline Noakes, RSTS Professional U.K. Rep.*]

Have you any plans for future articles on WP in general or WORD-11 in particular? Surely someone out there has written some "bolt on goodies" for WORD-11 that he'd be prepared to publish. If you wish to use any material from WUGMAG please feel free to do so.

Keep up the good work; by the way, I thought your DECUS Commercial SIG seminar in London last autumn (sorry "fall") was the best day out of 1981! When's the next one?

Arthur J. Davis
Watson Hawksley, Terriers House,
Amersham Road, High Wycombe,
Buckinghamshire HP13 5AJ, England

If you hurry, you can catch us in London on June 8.

. . . continued on page 78

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

You should add DOPTER!

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

INCREASES SYSTEM PERFORMANCE UP TO 50%.

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

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

For More Information

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

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

SPH

System Performance House, Inc.

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

CIRCLE 108 ON READER CARD

BASIC MEMORY EXERCISING PROGRAMS

By R.A. Smith, New Jersey District Support, Digital Equipment Corporation

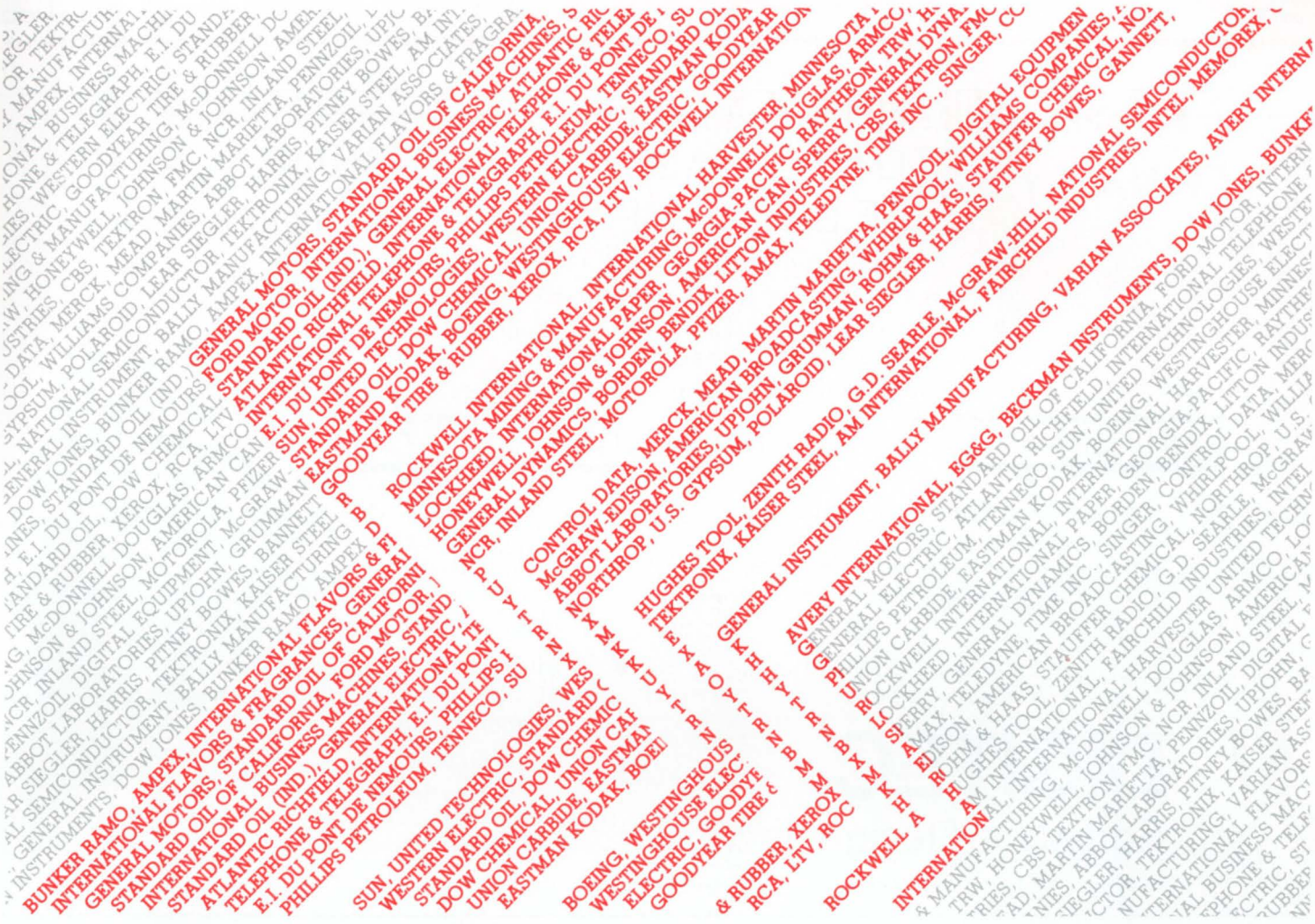
This is a simple basic plus memory test. To really get the most out of this particular test, load memory with as many of these jobs as needed to start swapping in and out of memory. Don't forget to leave room for your users! The error logger will log the errors. (Note: It is possible that your system will crash due to memory parity errors.) The memory will be tested to its full extent. The programs will run at priority -80 and runburst of 3 so when other jobs need to run, they will have priority over the background memory jobs.

```
1       PRINT 'THIS IS A SIMPLE MEMORY TEST'                &
\       PRINT 'TO GET BEST EFFECT, LOAD MEMORY'             &
\       PRINT 'TO THE POINT OF SWAPPING'                    &
\       PRINT 'RUNS AT PRIORITY -80 AND RUNBURST 3'         &
\       PRINT                                               &
\       PRINT 'WRITTEN BY. R.A. SMITH'                      &
\       PRINT 'NEW JERSEY DISTRICT'                         &
\       PRINT 'SJ X330'                                     &
\       PRINT 'DIGITAL EQUIPMENT CORP.'                     &
\       PRINT 'USE AT YOUR OWN RISK....'                    &

10      EXTEND

30      PRINT 'THIS PROGRAM IS DETACHING.....'                          &
\       DETACH.$ = SYS(CHR$(6%)+CHR$(7%))                               &
                                    ! DETACH JOB SYS CALL               &

50      PRIOR.RUNB$ =   SYS(CHR$(6%) +                                  &
                        CHR$(-13%) +                                    &
                        CHR$(255%) +                                    &
                        CHR$(-1%) +                                     &
                        CHR$(-80%) +                                    &
                        CHR$(-1%) +                                     &
                        CHR$(3%) +                                      &
                        CHR$(0%) +                                      &
                        CHR$(0%))                                       &
                                    ! SET PRIORITY TO -80 AND           &
                                    ! SET RUNBURST TO 3.                &

100     DIM A%(7000%) = 2%, B%(7000%) = 2%                              &
                                    ! MEMORY TEST SPACE                 &

300     A%(I%) = INT(RND(X)*32767%)     FOR I% = 1% TO 7000%            &
\       B%(I%) = A%(I%)                 FOR I% = 1% TO 7000%            &
                                        !THRASH MEMORY BY EQUATING      &
                                        ! A% = B%                       &

400     IF      A%(I%) <> B%(I%)                                        &
        THEN    PRINT   'I'; I%,                                        &
                        'A%'; A%(I%),                                   &
                        'B%'; B%(I%)                                    &
        FOR     I% = 1% TO 7000%                                        &
                                    ! COMPARE A'S WITH B'S.             &
                                    ! WILL HIBERNATE IF DO NOT          &
                                    ! COMPARE.                          &

500     GOTO 300                                                        &

32000   END
```

The following is an ATPK command file that has all that is needed to install this exerciser. Also enclosed is the ATPK log file of the actual building of this exerciser.

```
SW BP2COM
PIP MEMCOM.MAC = KB:
.TITLE MEMCOM
.PSECT MEMDAT,D,RW,GBL,REL,OVR
.BLKB 8192.
.END
^Z

RUN $MAC
MEMCOM=MEMCOM
^Z

PIP MEMCOM.CMD = KB:
SY:MEMCOM/-HD/PI,MEMCOM/SH,MEMCOM=SY:MEMCOM
/
PAR=TEST:0:0
STACK=0
//
^Z

TKB @MEMCOM

RUN $MAKSIL
MEMCOM
PIP MEMEXR.B2S = KB:
10      EXTEND
20      ON ERROR GOTO 19000
```

```
50      PRINT   '  THIS IS A TEST FOR MEMORY IT RUNS DETACHED   '       &
\       PRINT   'AND AT PRIORITY -80 & RUNBURST 1% SO NOT TO SLOW'      &
\       PRINT   '    THE SYSTEM WHEN USERS REALLY NEEDED IT.    '       &
\       PRINT   '  A RESIDENT LIBRARY, MEMCOM.LIB, WITH PSECT   '       &
\       PRINT   'MEMDAT IS USED FOR THE RELOCATION THROUGHOUT MEMORY.'  &
\       PRINT                                                           &
\       PRINT   '    CREATED MARCH 1982 BY R. A. SMITH          '       &
\       PRINT   '      DIGITAL EQUIPMENT CORPORATION            '       &
\       PRINT   '       N.J. DISTRICT SUPPORT SJ X330           '       &
\       PRINT           FOR P% = 1% TO 3%                               &

100     INPUT 'LOW MEMORY LIMIT<LOWEST NON RESIDENT MEMORY>'; LO.LIM%   &
\       INPUT 'HIGH MEMORY LIMIT<MAX MEMORY>'; HI.LIM%  &
\       HI.LIM% = 508%          IF HI.LIM% = 0%                         &
                                OR HI.LIM% < LO.LIM%                    &
\       INPUT   'NO. OF PASSES PER 4K SECTION <1>'; PASS.CNT%           &
\       PASS.CNT% = 1%          IF PASS.CNT% < 1%                       &
\       INPUT   'OPTIONAL DATA PATTERN <RANDOM>'; DATA.%                &
\       PRINT   'TESTING MEMORY LOCATIONS ';                            &
                LO.LIM%; 'K TO ';                                       &
                HI.LIM%; 'K. '                                          &
                                    ! GET MEMORY LIMITS AND             &
                                    ! NO. OF PASSES/ 4K SECTION         &
                                    ! MAXIMUM MEMORY SHOULD BE SET      &
                                    ! MAX MEM SIZE - 4%                 &
                                    ! IN STATEMENT 3 FOR HI LIMIT.      &

150     PRINT   'DETACHING ............'                                 &
\       S$ = SYS(CHR$(6%)+CHR$(7%))                                     &
                                    !DETACH SYS CALL                    &

170     PROIR.$ =       SYS(CHR$(6%) +                                  &
                        CHR$(-13%) +                                    &
                        CHR$(255%) +                                    &
                        CHR$(-1%) +                                     &
                        CHR$(-80%) +                                    &
                        CHR$(-1%) +                                     &
                        CHR$(1%))                                       &
                                    ! SET PRIORITY TO -80               &
                                    ! SET RUNBURST TO 1                 &

200     FOR     MEM.PLC% = HI.LIM% TO LO.LIM% STEP -4%                  &
                                    ! START AT HICORE AND               &
                                    ! WORK DOWN 4K INCREMENTS           &

300             DIM NAME.%(30%)                                         &
\               NAME.%(Z%) = 0% FOR Z% = 0% TO 30%                      &
\               LIB.NAM$ = 'SY:[1,80]MEMCOM.LIB'                        &
\               CHANGE SYS(CHR$(6%)+CHR$(-10%)+LIB.NAM$) TO NAME.%      &
\               NAME.%(0%) = 30%                                        &
\               NAME.%(1%) = 6%                                         &
\               NAME.%(2%) = -18%                                       &
\               NAME.%(3%) = 22%                                        &
\               NAME.%(11%) = MEM.PLC%                                  &
\               NAME.%(12%) = SWAP%(MEM.PLC%)                           &
\               NAME.%(18%) = 0%                                        &
\               CHANGE NAME.% TO NAME.$                                 &
\               S%    SYS(NAME.$)                                       &
                                            !UNLOAD RES LIBRARY         &

400             CHANGE SYS(CHR$(6%)+CHR$(-10%)+LIB.NAM$) TO NAME.%      &
\               NAME.%(0%) = 30%                                        &
\               NAME.%(1%) = 6%                                         &
\               NAME.%(2%) = -18%                                       &
\               NAME.%(3%) = 18%                                        &
\               NAME.%(11%) = MEM.PLC%                                  &
\               NAME.%(12%) = SWAP%(MEM.PLC%)                           &
\               NAME.%(18%) = 0%                                        &
\               CHANGE NAME.% TO NAME.$                                 &
\               S$ = SYS(NAME.$)                                        &
                                            !LOAD RESIDENT LIBRARY      &
                                            ! AT NEW LOCATION           &

500             CALL    MEMUSE(PASS.CNT%,DATA.%)                        &
                                            !GO TEST MEMORY             &

600     NEXT    MEM.PLC%                                                &
\       GOTO 200                                                        &
                                            !START OVER AFTER           &
                                            ! SPECIFIED MEMORY          &
                                            ! RANGE HAS BEEN TESTED.    &

19000   !*****************************&
        !                             &
        !       ERROR ROUTINE         &
        !                             &
        !*****************************&

19100   IF      ERL = 400                                               &
        THEN    RESUME 600                                              &
                                            ! IGNORE ERRORS CAUSED      &
                                            ! BY TRYING MEMORY          &
                                            ! ALREADY ALLOCATED TO      &
                                            ! RESIDENT LIBRARIES        &
                                            ! RTS AND MONITOR           &

19200   PRINT   'ERROR';ERR                                             &
\       ON ERROR GOTO 0                                                 &

32767   END
^Z

PIP MEMUSE.B2S = KB:
10      SUB     MEMUSE(PASS.CNT%,DATA.%)                                &
\       EXTEND
100     COM(MEMDAT)     WORD$(4095%)
                                    ! 4K OF RES. LIBRARY                &
```



Fortune 500's finest have selected us for use with their DEC and DG minicomputers.

For over a decade, System Industries has provided reliable and cost-effective data base storage alternatives to many Fortune 500 companies using DEC and DG minicomputers. SI has over 20,000 installations around the world today, because we've given DEC and DG users just what they wanted:

1. MORE RELIABILITY. SI provides great data base solutions—like our redundant Add-On Non-Stop Reliability (ANSR) systems. Or our 160 or 675 Mbyte, SMD-compatible Winchester storage systems, for more dependability and increased throughput.

2. SPACE SAVING CONFIGURATIONS. The SI storage alternative puts many more Mbytes in a single footprint. We'll give you the equivalent of 8 "logical" RM03s in the space normally occupied by one RM03 from DEC. Or the equivalent of 6 RM05s in the space normally occupied by four from DEC.

3. EXCELLENT SERVICE. We maintain service centers in the U.S. and Europe. We offer 4-hour response to calls within a 50-mile radius of major U.S. cities. And 24 hours/day, 7 days/week service contracts in all metropolitan areas.

4. FAST DELIVERY. SI has built its reputation on quick delivery. Typically, we can give DEC and DG users value-packed data base storage systems in 30 to 45 days ARO.

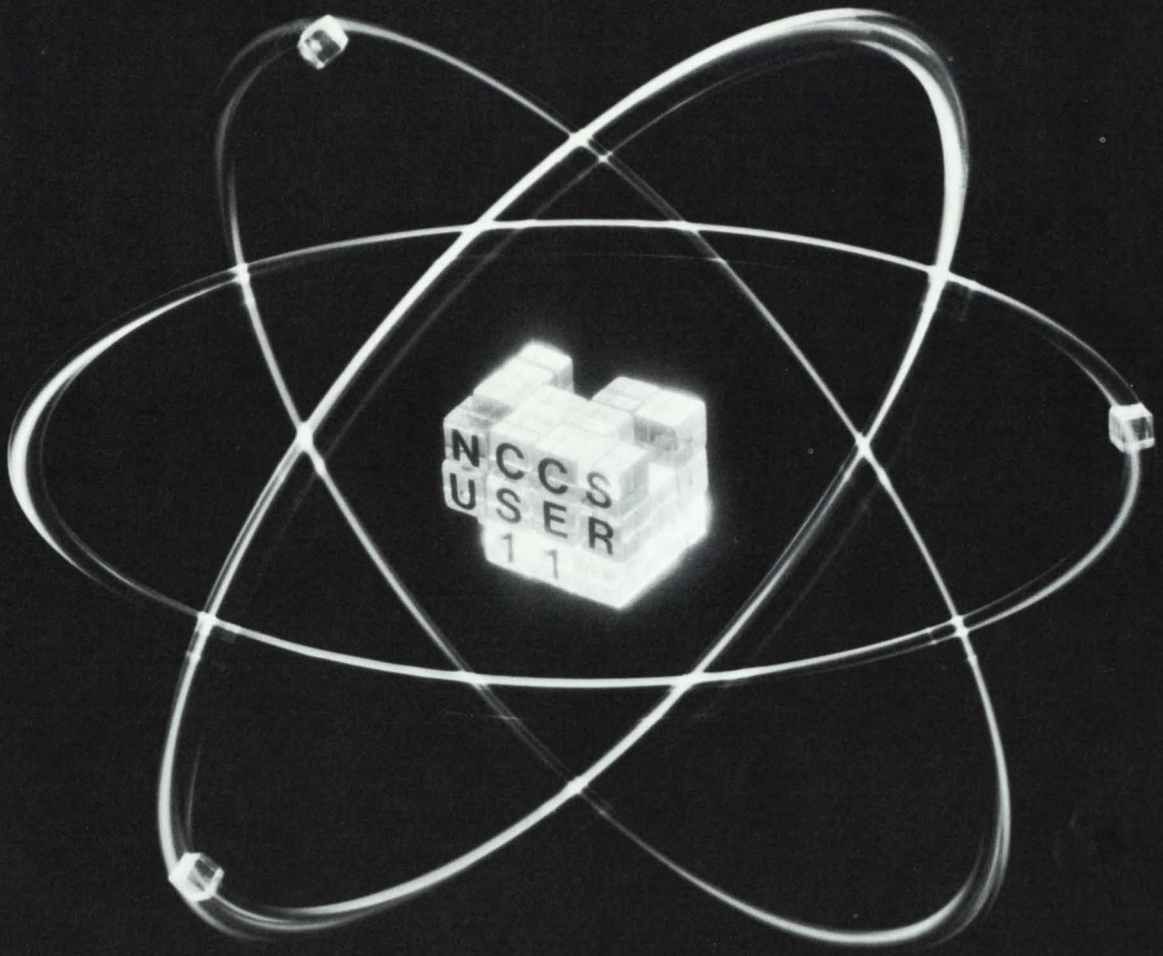
For more information about the popular choice of the Fortune 500, and the accepted storage alternative of DEC and DG minicomputer users, contact us today at any of the offices listed below.

System Industries
the only viable alternative

United States: 1855 Barber Lane, Milpitas, CA 95035, (408) 942-1212, Telex 346459.
Europe: System Industries (Europe), System House, Guildford Road, Woking, Surrey, GU22 7QQ, England, (44) 4862 5077, Telex 859124.
California (714) 851-6289, (213) 822-3161; **Colorado** (303) 986-1591; **Georgia** (404) 955-2252; **Illinois** (312) 948-9330; **Massachusetts** (617) 431-7240; **Michigan** (313) 663-4925; **New Jersey** (201) 839-8650; **New York** (212) 696-0650; **Ohio** (513) 771-0075; **Texas** (713) 497-7224, (214) 386-8776; **Virginia** (703) 734-9700; **Washington** (206) 451-8791; **West Germany** (49) 6102 5464; **Sweden** (46) 08-63 16 20

CIRCLE 20 ON READER CARD

```
200     FOR     TIMES.% = 1% TO PASS.CNT%                               &
\               IF      DATA.%                                          &
                THEN    GD.DATA% = DATA.%                               &
                ELSE    GD.DATA% = INT(RND * 32767%)                    &
                                        !USE SPECIFIED PATTERN          &
                                        !UNLESS NONE SPECIFIED          &

300             WORD%(I%) = GD.DATA%    FOR I% = 0% TO 4095%            &
                                        !FILL LIBRARY WITH              &
                                        !DATA PATTERN                   &

400             FOR     WRD.CNT% = 0% TO 4095%                          &
\               BD.DATA% = WORD%(WRD.CNT%)                              &
\               IF      GD.DATA% <> BD.DATA%                            &
                THEN    PRINT 'DATA MISCOMPARE'                         &
\                       PRINT 'GOOD DATA: ';GD.DATA%                    &
\               PRINT 'BAD DATA: ';BD.DATA%                             &

450             NEXT    WRD.CNT%                                        &
                                        !DO ALL 4096 WORDS              &

500     NEXT TIMES.%                                                    &
                                        !DO FOR NO. OF PASSES           &
                                        ! SPECIFIED.                    &


32000  SUBEND
^Z

OLD MEMUSE.B2S
COM/OBJ
OLD MEMEXR.B2S
COM/OBJ

PIP MEMEXR.ODL = KB:
        .ROOT USER
USER:   .FCTR SY:MEMEXR-LIBR-*(MEMUSE-LIBR)
LIBR:   .FCTR LB:BP2COM/LB
        .END
^Z

PIP MEMEXR.CMD = KB:
SY:MEMEXR=SY:MEMEXR/MP
HISEG=BP2COM
UNITS = 12
ASG = SY:5:6:7:8:9:10:11:12
EXTTSK= 512
//
^Z

TKB @MEMEXR

UT ADD LIBRARY [1,80]MEMCOM<0>/RW/ADDR:508

RUN MEMEXR
0
508
50

Ready

SW BP2COM


BASIC2

PIP MEMCOM.MAC = KB:
.TITLE MEMCOM
.PSECT MEMDAT,D,RW,GBL,REL,OVR
.BLKB 8192.
.END
^Z


BASIC2


RUN $MAC
MAC>MEMCOM=MEMCOM

MAC>^Z


BASIC2


PIP MEMCOM.CMD = KB:
SY:MEMCOM/-HD/PI,MEMCOM/SH,MEMCOM=SY:MEMCOM
/
PAR=TEST:0:0
STACK=0
//
^Z


BASIC2


TKB @MEMCOM


BASIC2


RUN $MAKSIL
MAKSIL V7.0-07 RSTS V7.0-07 NJDIST
Resident Library name? MEMCOM
Task-built Resident Library input file <MEMCOM.TSK>?
Include symbol table (Yes/No) <Yes>?
```



USER-11: POWERFULLY PRODUCTIVE.

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

USER-11 is a high-performance database system.

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

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

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

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

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

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

NCCS USER 0110

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

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

CIRCLE 30 ON READER CARD

```
Symbol table input file <MEMCOM.STB>?
Resident Library output file <MEMCOM.LIB>?
MEMCOM built in 4 K-words, 0 symbols in the directory
MEMCOM.TSK renamed to MEMCOM.TSK<40>

BASIC2

RUN MEMEXR
  THIS IS A TEST FOR MEMORY IT RUNS DETACHED
AND AT PRIORITY -80 RUNBURST 1% SO NOT TO SLOW
    THE SYSTEM WHEN USERS REALLY NEEDED IT.
  A RESIDENT LIBRARY, MEMCOM.LIB, WITH PSECT
MEMDAT IS USED FOR THE RELOCATION THROUGHOUT MEMORY.

    CREATED MARCH 1982 BY R. A. SMITH
      DIGITAL EQUIPMENT CORPORATION
        N.J. DISTRICT SUPPORT SJ X330


LOW MEMEORY LIMIT<LOWEST NON RESIDENT MEMORY>? 0
HIGH MEMORY LIMIT<MAX MEMORY>? 508
NO. OF PASSES PER 4K SECTION <1>? 50
OPTIONAL DATA PATTERN <RANDOM>?
TESTING MEMORY LOACTIONS  0 K TO  508 K.
DETACHING ............
```

MINICOMPUTER PROGRAM ERRORS DUE TO PARASTATIC CONDUCTANCE

By D.A. Lowe, Staff Assistant, Occidental Life of California

Extensive research in the Home Office has determined that a large percentage of minicomputer system program errors are being caused by parastatic conductance resulting from differential spurving of the hydroscoptic marselvanes located in the prefabulated amilite base of the unilateral detractor mechanism.

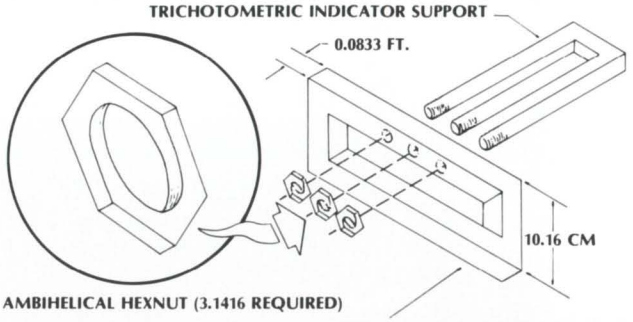
Digital Equipment Corporation has been unable to offer any remedy for this troublesome situation other than to recommend manestically spacing the grouting brushes on the periphery of the nubbing purwell.

Although on the surface this would appear to alleviate the problem, we have found that this leads to further complications causing the regurgitative wennel sprocket to transmit microgriffage to the anhydrous dangling pin, from whence it is modulated, amplified, and splitnagled, thus causing transendental hopper dadoscope failure. This, in turn, causes quasipiestic depleneration of the bitumogeneous sprandels, thus leading to an even higher level of high RMP peak nivel-sheave voltage which further magnifies the amnesial slump.

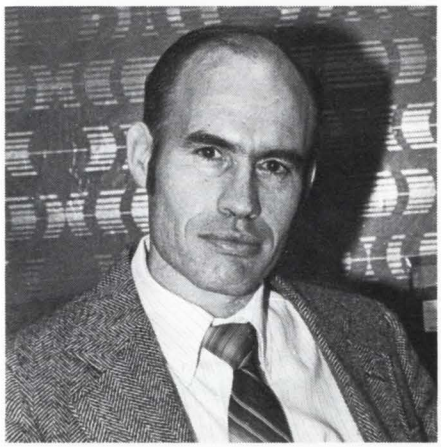
It should be apparent that any successful solution has to be based on the regeneration of low-ohmic nofers combined with a high degree of medial interation of magneto-reluctance and resistance to atmospherical rillarah.

Fortunately, we have discovered a simple and effective remedy which involves merely modifying the spiral decommutator with the installation of a rectabular extrusion bracket and trichotometric indicator support (see attachment).* These items should be purchased (out of petty cash) from any local supply house and installed immediately. Upon installation, the above cited malfunctions should be reduced significantly and you should experience greatly increased non-reversible tremic amifacience.

*Note that the special ambihelical hexnut is unique in that any attempt to remove it in the conventional manner only succeeds in tightening it. Because of this design, the nut must be fully screwed on before it can be screwed off.



"Ambase is making our lives easier..."



Sam Walden,
Vernay Laboratories, Inc.,
Yellow Springs, Ohio,
manufacturer of laboratory gaskets,
using AMCOR'S AMBASE and
AMFACS systems in an integrated
environment.

"We are using AMBASE for a variety of things, specs and standards on all parts and types of tools for each division, mailing list, order entry and tool masters. We also have a program directory which tells the user what programs are available.

AMBASE is making our lives easier through aid messages, and more reliable data. It puts the data in a common area instead of different ones. We are rewriting all of our programs under AMBASE.

Amcor's training facilities are very conducive to learning and the training program complete. Installation of AMBASE was very, very, easy, and Amcor's support for AMBASE is good. Everything has been taken care of quickly.

Using AMBASE in an integrated environment with Amcor's application products is advantageous to us because the systems have a number of things built into them that would have taken us a long time to develop. Getting everything under AMBASE completely will bring about programming conformity."

AMBASE is a revolutionary state-of-the-art system for application development and data base management. AMBASE is increasing programming productivity worldwide from 100-900%. In addition to the data management system, AMBASE includes a report generator, query language, screen formatter and automatic code generator.

If you would like to find out more about AMBASE, just clip and mail the coupon, or give us a call, TOLL FREE at 1-800-626-6268. We will send you free information immediately.

Mail to:
AMCOR Computer Corp.
1900 Plantside Drive
Louisville, Kentucky 40299

Please send more information on AMBASE DEC RSTS Software to:

Name

Company

Address

City

State Zip

Phone

amcor computer corp.
a Kaneb company

Headquarters: 1900 Plantside Dr., Louisville, KY 40299 Regional Offices: Nationwide

CIRCLE 3 ON READER CARD

EDITING SYSTEMS

By David Spencer, Infinity Software Corporation

TALKING WITH THE WORLD IN TECO

For the last two issues I have discussed EDT editing techniques. In this issue I'm going to go back to the bread and butter, so to speak. Here are a few more tips on making better use of TECO.

(If you're not a TECO expert, don't be frightened off. Just get out your TECO pocket guide or the "PDP-11 TECO User's Guide" that comes with the RSTS/E documentation set. Whenever you see something here you don't recognize, stop and look it up! Either document does a good job of explaining TECO editing commands. It may take a little longer to read this article, but TECO is really quite easy to understand.)

Writing code in TECO is just like any other new language. You really don't feel like you've done much until something goes in or comes out. A REAL program accepts something from the keyboard and DOES something with it.

So let's build a simple program that accepts input from the keyboard and does editing with it.

1.0 What kind of program?

Good question. Fortunately, I have something in mind.

Datatrieve is a wonderful tool. You can write reports with it all day. But have you ever printed one of those reports on a slow printer like an LA180? Yep, lots and lots of trailing spaces. A TECO program would be just the thing to take those useless spaces off.

2.0 Define the application

Now that we have an idea what we want to do, let's write it down.

"This should be a program that you can run, ask for a filename, remove those ugly spaces, and then return to the keyboard monitor."

Processing steps:

1. Print a prompt
2. Input filename
3. Open filename
4. For entire file; replace all spaces + CR with CR
5. Close file
6. Exit

3.0 Build the program

Above, I've listed the steps to perform the de-spacing. If you've noticed, this list is not greatly detailed. That's okay because for now we're only on a high level. I have no doubt that getting input and opening a file will both require a number of steps by themselves. We'll come to that level of detail later.

The first item on the list is to print a prompt. The TECO command Control/A can be used to print both the header and the prompt at the same time.

```
@↑A/UNFILL        Remove trailing spaces from a file
File to squish? /
```

This program will be called "UNFILL" (because it removes filler). The one line description in the header is a good idea because it's likely in the future I'll forget what UNFILL does. Notice that the prompt is left dangling so it will look like any other normal input prompt.

4.0 A TECO programming hint

By the way, I always use the indirect version of TECO commands. It is very difficult for most editors to handle typing of control and escape characters without interpreting them as you hadn't intended. If you really need speed, use the SQU utility from the TECO distribution to compress a copy of your program later.

5.0 Writing a keyboard input routine

Like the main program, it would be a good idea to define what an input routine is supposed to do.

First, it's probably a good bet to assume the terminal will be a scope. This allows rubbing out characters with a backspace, space, backspace combination.

The TECO manual reveals that TECO accepts input on a character by character basis. So, our code must be tailored for this situation.

Keyboard input is normally terminated with a carriage-return. The input routine must complete on that character. And last, when input is complete, the input string should be left somewhere for use by following routines.

Input routine steps:

```
  Until carriage-return detected;
    Get a character
    If <delete> then delete character (if present)
    otherwise
    insert character
End-Loop
(leave text in editing buffer)
```

The following routine will perform the input. Keep your hat on, this is actually a very simple and structured piece of code.

```
<
! Read a character and store in numeric A !
 ↑TUA
! If it's <del> then !
 QA-127"=
!    if no characters in the buffer beep !
        Z"=
            7↑T
! else !
             |
! delete last character, scope rubout !
                -D 8↑T 32↑T 8Z↑T
                '
! Get another character !
          F<
           '
```

```
! If it's a <cr> then !
  QA-13" =
! Suck up the dangling <lf> and end !
        ↑TUA 0;

! Otherwise insert the character !
  QA@I//
>
```

6.0 Code to open the given file

Some code is required to open the filename after is it supplied. To do this, we'll use one of the neat features of TECO, the ability to create code on the fly and then execute it.

The file to open has been left sitting in the buffer. If a few characters are inserted around that filename, it can be made into a TECO command. That text can be placed in a register with the "X" command. (A register is just a string in BASIC.) The "M" command can then be used to execute the text stored in that register.

Just go through the following steps, and how this happens should become more clear.

```
! Go to the top of the buffer !
J
! Insert edit both command !
@I/EB/
! Go to the end of the buffer !
ZJ
! Insert an escape !
27@I//
! Stuff created command in register A !
HXA
! Remove command from buffer !
HK
! Execute command to open file !
MA
! Bring in first page !
P
```

After inserting the edit with backup command ("EB") and the terminating escape, the text in the buffer would look something like this: "EBfile<esc>". The reason the buffer is cleared before reading in the first page is to prevent the filename text from getting into the report file and/or generating nasty errors.

7.0 Edit the file

The following is a simple command that will search for a string of spaces followed by a carriage-return. If it is found, then replace it with a simple carriage-return.

```
<
@FN/↑ES
/
/;
>
```

8.0 Finishing up

After the search and replace is complete, our task is done. All that's left to do is to exit TECO. This is done with the "EX" command.

```
EX
<esc><esc>
```
(Pretend "<esc>" is a real escape character.)

9.0 Little details

There are a few extras required in any TECO program. Not much is said about them, but they are required to get the job done.

The first falls under the "but everybody knows that!" category. Put simply, end your program with two escapes. The escapes tell TECO that your program is complete. If you don't put them in, TECO will look at your program until all characters are read. When TECO exhausts the input file (your program), it switches to looking to keyboard for more input. At this point, it would be possible to type two escape characters, and the program would start normally. This, however, is considered poor form.

Another of these items is to turn off the "EI" command. If this isn't done, your program won't operate properly. What happens is when TECO loads your program, it does it in a way that simulates somebody typing it from a keyboard. After the whole program is loaded (two escapes found), TECO continues to look at the file it reads (your program) for all keyboard input. This usually results in your code bombing from strange input. In order to make TECO switch to the keyboard for input, an "EI" with no file specification will do the job.

10.0 Review the program

Now that UNFILL works, I've noticed that Datatrieve starts each report with a form-feed. Since I use a spooler, this is wasted paper. So let's add a statement to take that form-feed off.

In TECO, a page is terminated either by filling three quarters of the buffer or by encountering a form-feed. If it is a form-feed, TECO sets the Control/E flag and no form-feed is inserted into the buffer. Later, when TECO writes out a page, Control/E is used to tell when to write a form-feed on the output file.

In our case of Datatrieve reports, the first buffer should be blank because the first character in the file is a form-feed. An easy check is to test the buffer character count flag "Z". If the page is blank, then the append "A" command can be used to attach the next page to this one. One of the append command features is that it ignores the Control/E flag. That will remove the unwanted top-of-form quickly and easily.

```
! Go to the top of the buffer !
J
! Insert edit both command !
@I/EB/
! Go to the end of the buffer !
ZJ
! Insert an escape !
27@I//
! Stuff created command in register A !
HXA
! Remove command from buffer !
HK
! Execute command to open file !
MA
! Bring in first page !
P
! *** NEW CODE *** !
! If the page is blank then append the next one !
Z"= A '
```

11.0 Final program listing

Here is the final program listing. Most of the comments have been removed for "clarity".

```
! Macro to remove trailing spaces and other nasties!

! Author: David Spencer                                    !
! Infinity Software Corporation                            !

! This routine requires the following registers:           !
! A$      Filename to open                                 !
! A%      Input text character                             !

@EI//

@↑A/UNFILL      Remove trailing spaces from a file

File to squish? /
<

  QA-127"=
          Z"=
              7↑T
               |
              -D 8↑T 32↑T 8↑T
             '
          F<
        '

  QA-13"=
           ↑TUA 0;
        '

  QA@I//
>

J
@I/EB/
ZJ
27@I//
HXA
HK
MA
P
Z"= A '

<
@FN/↑ES
/
/:
>

EX
<esc><esc>
```

That's all there is to it. Armed with your new found knowledge, you should be able to write many useful TECO programs! Until next time, have a good exit.

EMULEX ANNOUNCES 34 NEW DISK SUBSYSTEMS FOR DEC USERS.

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

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

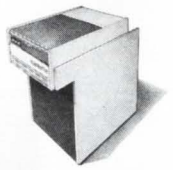
Emulex is known worldwide as *the* DEC controller expert. Over 7500 delivered. So we're a natural to provide you with the widest choice of complete disk subsystems you've ever seen.

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

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

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

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



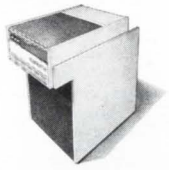
For LSI-11/23 Q-BUS: 80 MB Removable Emulates DEC RM02/03

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

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

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

For VAX UNIBUS: 160 MB Fixed Emulates DEC RM03 Expanded



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

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

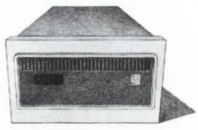
For LSI-11/23 Q-BUS: 96 MB Fixed/Removable Emulates DEC RK06/07

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

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

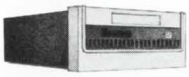
For VAX UNIBUS: 675 MB Fixed Emulates DEC RM05 Expanded

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



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

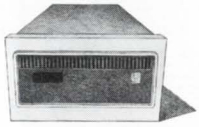
For PDP-11/04-60 UNIBUS: 80 MB Removable Emulates DEC RM02/03



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

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

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



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

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

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

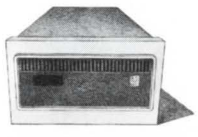
For PDP-11/04-60 UNIBUS: 96 MB Fixed/Removable Emulates DEC RK06/07

For VAX UNIBUS: 80 MB Removable Emulates DEC RM03

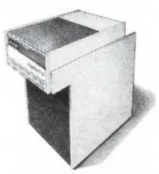
For VAX UNIBUS: 64 MB Fixed/Removable Emulates DEC RK07

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

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



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



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

For VAX UNIBUS: 300 MB Removable Emulates DEC RM05

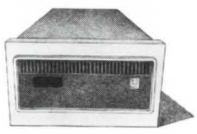
For VAX UNIBUS: 96 MB Fixed/Removable Emulates DEC RK07

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

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

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

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



For VAX UNIBUS: 80 MB Fixed Emulates DEC RM03

EMULEX

THE GENUINE ALTERNATIVE.

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

CIRCLE 58 ON READER CARD

CCLMAN
CCL Manager For RSTS/E

1.0 Description and Concept

CCLMAN is a very useful program written in Basic-Plus 2 allowing the system manager to conserve small buffers by not ADDing many, many CCL commands to the RSTS/E small buffer pool. This program will store these extra CCL commands in an RMS-indexed file, then they are executed by typing '@@ <Command>' instead of just '<Command>'.

As distributed, over 4000 CCL commands can be stored in the CCL file. The system manager may Add, Remove, or List the CCLs currently stored anytime during normal timesharing.

The CCLMAN program requires only 2 CCLs itself to execute correctly. Note that certain CCL commands MUST remain in the normal 'UTILTY' CCL structure to allow items supplied by DEC to execute correctly (Such as AUTOPATCH, SYSTEM GENERATION, LANGUAGE GENERATION, etc.). CCLs may be added to this special CCL library and be in the standard CCL structure AT THE SAME TIME. If this is done, these 'doubly defined' CCLs will execute either way, normal or through CCLMAN.

The format to Enter or Remove CCLs to the CCLMAN structure is the exact same as used to enter CCLs to UTILTY.

It's format follows:

```
#ADD XYZ-TEST=DEV:[Acct]PROGRM.EXT;[PRIV] LNUM
    -or-
#REMOVE XYZ-TEST=DEV:[Acct]PROGRM.EXT;[PRIV] LNUM
```

NOTE ON ADDing a CCL, the 'DEV:[Acct]' is optional and on removing a CCL, everything past the '=' is optional.

Commands may also be abbreviated to 1 letter, so 'A' or 'AD' or 'ADD' are functionally equivalent.

2.0 Installation

CCLMAN is written in Basic-Plus 2, installation is as follows:

```
BP2
Basic2
OLD CCLMAN
Basic2
COMPILE/OBJ/LINE/CHAIN
Basic2
BUILD/IND
Basic2
TKB @CCLMAN
Ready
```

At this time CCLMAN.TSK should be moved to the account it will reside in and MUST have a protection code of <232>.

As mentioned before, two CCLSs are required to allow CCLMAN to execute properly, they follow:

```
#ADD @@-@@@@@@@@=[Acct]CCLMAN.TSK;PRIV 30000
#ADD CCL-MAN=[Acct]CCLMAN.TSK;PRIV 30000
```

NOTE, THE CCL '@@' will probably have to be added before the required DEC CCL '@' which is used for ATPK or UTILTY will give you a 'Name or Device in Use' error.

Note, the first time CCLMAN executes, the file 'CCL.DAT' will be created in the ACCOUNT that CCLMAN resides in.

Installation is now complete!!!

3.0 Usage and Examples

Adding, Removing, or Listing CCLs can now be entered as a normal CCL call or by 'Running' the program, CCLMAN, directly. Examples follow:

```
CCL ADD TEST-CCL=[1,4]TEST.TSK;PRIV 6655

Ready

RUN [Acct]CCLMAN
CCLMAN V7.0-07      RSTS V7.0-07 Installation Name
#ADD TEST-CCL=[1,4]TEST.TSK;PRIV 6655
#.Z

Ready
```

The above two commands are equivalent.

3.1 Commands and their Formats

A list of commands and their formats follows:

```
A[DD] TEST-CCL=[DEV:(ACCT)]PROGRM.[*]/[EXT]:[PRIV] LNUM
  Add CCL to the CCL structure.
R[EMOVE] TEST-CCL=
  Remove a CCL from the CCL structure.
C[LEAR]
  Clear complete CCL table. BEWARE!!
  You will be prompted for confirmation
H[elp]
  Help explanation for CCLs in structure.
L[IST]
  List CCLs in structure, by alphabetical format.
@FILENM.CMD
  Indirectly execute a file of commands, prompt during execution is '*'.
```

4.0 Questions or Comments

Questions about the use of this program may be directed to myself at the following address: PHILIP HUNT, C/O O.L.F.B.P., 6400 E. BROAD STREET, COLUMBUS, OH 43213, (614)863-3473

A tape of this program is available if you send $15.00 and a tape to the above address. Specifiy whether you want 800 or 1600 BPI and whether you want ANSI or DOS format tape.

. . . continued on page 28

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

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

More Flexibility.

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

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

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

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

More Control.

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

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

More Productivity.

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

More Support.

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

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

MAS-M

The Software System That Can Help You Do More.

MARTIN MARIETTA DATA SYSTEMS

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

MARTIN MARIETTA

CIRCLE 111 ON READER CARD

"FROM ALL ENEMIES, FOREIGN AND DOMESTIC"

PROTECTING A RSTS SYSTEM

By Richard Davis Mallery, The RSTS Professional, PO Box 361, Fort Washington, PA 19034

INTRODUCTION

The subject of this paper is security, or should I say survival? The two go hand in hand, both in government and in our little world of computer systems. In a sense, a computer system is much like an island nation — surrounded on all sides by impassable obstacles and connected to the outside world by thin threads of communication. Many a Maginot line of physical defense has been constructed around computer systems only to have the system sacked and pillaged by a child with a telephone.

First I will define the terms in the title, and then attempt to define and limit the problem. The first noun is 'enemies'. By 'enemies', I mean anyone or anything that attempts access to part of your computer system or its communication network without authorization either intentionally or by accident.

'Foreign' implies someone that does not belong here. In that sense, I mean anyone logged out who should not be permitted to log in — anywhere. By 'domestic', I mean anyone logged in properly, or able to log into a given account or group. A 'domestic' becomes a domestic enemy when he, she or it attempts any form of unauthorized access. There is an implication here that if you are 'logged-out', nothing in the system is accessible to you except for the log-in procedure. That is seldom true in RSTS systems that have not been protected.

Granted that steps have been taken to insure the above premise, our problem is now somewhat neatly divided into two areas: logged-out and logged-in. If an enemy is logged out, you are successful if you keep him, her or it logged out. If a logged-in entity becomes an enemy by violating his 'space' in the system, we are successful if we keep him where he belongs, and deny him any information from another's or the system's 'space'.

Above and beyond the above rather limited definition of success, we serve our employer and society well if through our diligence and cleverness, a criminal is occasionally captured and punished.

LOGGED-OUT SECURITY

The only mode of access to a RSTS system is through a keyboard. To my knowledge, a tape drive has never logged in. This portion of the paper will discuss the vulnerabilities of keyboards.

There are really three kinds of keyboards: pseudo keyboards, keyboards connected to 'DL' type interfaces, and those connected to 'DH/DZ' interfaces. I will not discuss unsupported interfaces such as synchronous lines; interfaces that do not end up as keyboards in the RSTS internal sense.

Pseudo keyboards are never connected in the physical sense. They exist only as control blocks in the monitor, but other than their intangibility, they are real keyboards in every other sense. Commands 'forced' into their buffers are as real to RSTS as the characters that formed this paragraph. There are very few, if any systems that have no pseudo keyboards, and their location is always at the low end of the list, just above the DL type interfaces.

Physical (non-pseudo) keyboards, regardless of their interface have the added attribute of location. They have the ability to connect to the outside world. (Outside means beyond the interface connector.) A keyboard may be connected or not to a wide variety of devices, either through a simple local null-modem cable, or through some form of communication device.

In conducting a security audit, it is always an interesting exercise to list the keyboards that should be disconnected, and then determine that they are indeed disconnected. Many an interesting discovery has been made buried in the inevitable mess of cables behind a CPU.

There is only a small difference between 'DL' type interfaces and 'DH/DZ' types. The speeds of 'DL' lines cannot be changed by RSTS, but only by setting switches on the interface boards themselves. This is a liability when trying to shut off probing dial-in enemies.

In summary, all keyboards are at risk because an enemy can gain access (get logged-in) over any one of them. The physical location of the device connected to the interface port (kb) may have some effect on the accessibility of that keyboard, but any wire that passes from the computer to the terminal is subject to tapping, even though the ultimate destination is secure.

Keyboards connected to the dial or packet networks are perhaps the most vulnerable.

DIAL-IN VULNERABILITIES

The sudden emergence of the hobby computer has created a situation that can only be classified as a crisis. As the micros proliferated, dial-in bulletin boards and the like became popular. The new 'network' created by these 'information utilities' generated a market for modems. Now, no hobby machine is complete without one. In fact, there is one modem on the market that boasts of its auto-dial capacity. This modem can be used to scan an entire telephone exchange at a time, and a ten line basic program can produce a list of all computers that answered with the correct tone in a few short hours. At this point, our new enemy is free to probe all these numbers at leisure.

The last paragraph should put to rest all arguments about the merits of unlisted or rotated dial-in numbers. One

Productivity Plus

This is what TRIDAKIT offers. Productivity, plus Accuracy, plus More. Here's why.

TRIDAKIT

TRIDAKIT is a set of development tools that utilize the power of RSTS/E to its fullest. Included are documentation, programming, operations, and project management tools.

Incorporated in TRIDAKIT is a powerful File Manager which can open and maintain 255 files simultaneously. TRIDAKIT based applications allow for cacheing of files and reduced use of the RSTS/E File Processor (FIP), and a reduction in the use of small buffers which increases the speed of these applications while allowing normal operations to continue.

PRODUCTIVITY

Up to 90% of program code is generated automatically by TRIDAKIT. Programmers can instruct TRIDAKIT to produce whole libraries of code with simple easy to use commands.

Complete documentation is automatically produced as a by-product of application definition instead of "after-the-fact".

Changing the documentation changes the program code. Documentation is always current!

TRIDAKIT promotes "standards" for coding applications. This allows several programmers to work on one application. It also lends to easy maintainability.

ACCURACY

TRIDAKIT promotes the "team" approach to application development. With TRIDAKIT each team member always has the latest, most up-to-date, version of the documentation. This provides a clear definition of tasks to be completed and helps control application development.

PLUS MORE

Security is strictly enforced with TRIDAKIT. Each user of TRIDAKIT must be defined by operations before any application can be used. The user must then "log in" to the application that has been assigned. TRIDAKIT deals with special problems that are commonly skipped or forgotten in applications such as audit trails, I/O tracing, and "point-of-failure" data base recovery. These are automatically included in TRIDAKIT based applications.

INFORMATION

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

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

Tridacor Systems, Inc.
820 Freeway Drive North
Suite 211
Columbus, Ohio 43229
614-431-0805

of the weakest points of RSTS is that there is no warning given to the system manager that a probe is going on, and no limit to the number of re-tries available to a determined enemy. At best, LOGIN can make it expensive by hanging up after each 'access-denied' message, but who cares if it is a local call and the modem does the dialing?

The second major vulnerability is perhaps characterized as "RSTS Chauvenism":

```
RSTS V7.0-01 ABC Manufacturing Co. Job 36 KB15: 15-Jan-82
15:36
#
```

The first thing we tell the enemy is our name, rank and serial number! No more need be said.

Once the enemy has identified the fact that we are a RSTS system, there is a wealth of information available as to the make-up of that system. Many 'innocent' facts are now of great use to the enemy. To wit:

1) The following accounts exist: [0,1], [1,1], [1,2] and others

2) These accounts almost certainly have passwords.

3) There is a high likelihood that one of these passwords is a 'ringer' like SYSLIB, etc.

4) If the phone doesn't hang up on ctrl/c, we are free to see what ccls will work logged-out (DIR and SYS/S, for instance). Both DIR and SYS will give us a list of the other ppns in use by this system.

5) After trying all the common 'ringer' passwords and accounts, the enemy programs his micro to try all the possible passwords for [1,2], one after another. So what if it takes all weekend? The enemy can snooze till the micro detects an answer other than 'Invalid entry — try again' and 'Access denied'.In an unprotected RSTS system, all of the above will proceed undetected, unless someone is watching via systat.

In summary, any dial-in line that uses a normal modem, is exposed to the infinite patience, thoroughness, and blinding speed of computerized probing which can proceed undetected for extended periods of time at almost no cost or effort to the enemy. In my opinion, then, any LOGIN that allows direct access to privileged accounts with only a single password is already compromised.

That concept should, however be extended. Leaving behind the hobby 'enemy', we are also vulnerable to similar, but more determined efforts to break in over any 'physical' keyboard on our system. There is no wire that cannot be spliced, no phone line that does not terminate somewhere, no lock too good for a professional criminal. We are all equally exposed to the hobby criminal and phone phreak simply by the connection of a single line to the dial network. Our exposure to the dedicated professional is probably measured by the real or apparent value of the information available on our system. We cannot consider application-internal security here, but should involve ourselves with the likelihood of penetration by highly motivated thieves.

Wire-tapping a leased or dial-up line will produce a full record of all transactions, including logins and security measures. The only known defense against the criminal tap is encryptation. Another thin defense is the imposition of time of day and day of week limitations, since most criminals will utilize their new-found passwords only after hours to prevent detection. Many encryptation techniques exist, and range in effectiveness from trivial to impenetrable.

LOGGED-IN VULNERABILITIES

Now we will address the problem of logged-in enemies: users of our system who have developed an appetite for information outside of their 'space' (a specific non-privileged account, perhaps only during working hours on Monday through Friday). The motivation may range from curiosity to larceny, but the object is the same. Often, passwords are written on terminals, yelled across rooms or become public domain by other means. You cannot guarantee the privacy of non-privileged passwords in an office environment. In fact, the more often you change them, the more often they are written down or passed verbally.

A strategy for limiting non-privileged access might be to limit specific terminals to specific accounts or account groups, whether or not the operator knows the correct password for other groups. In addition, no terminal should be allowed direct login access to privilege unless specifically and explicitly granted. In sum, the best solution to the password problem is to eliminate passwords as a direct factor in access. A terminal might be allowed into the 'accounts payable' and 'general ledger' accounts, but not into 'accounts receivable' or 'payroll'. In addition, none of these accesses would be allowed outside of regular hours or on weekends without system-manager intervention. In this case, you are relying on the presence of co-workers and supervisors to restrict excursions to unauthorized terminals. In no case, should any production accounts be directly accessible from dial-in lines, even though they are non-privileged.

There are many considerations for the system manager that impact logged-in behavior. I will list some here.

- Many systems have unnecessary CCLs that execute priv'd utilities. (attach, utility, . . .)
- Non privileged users can dismount disks.
- Dismounted disks, unless protected by a feature patch, can be opened NFS and the MFD dumped by any novice BASIC hack.
- Old copies of LOGIN.* and non-deleted copies of ACCT.SYS.
- Unprotected copies of DIR and SYSTAT and the like will give the enemy a bird's eye view of the system and its organization — when you are breaking passwords, it's nice to know the ppn to start on.

All of the above can be eliminated as 'penetration aids' by judicious use of feature patches and common sense.

DEFINING A STRATEGY

With the above as an introduction, I will attempt to reconstruct the processes that we followed in designing a security package for our RSTS system.

We chose a 'keyboard security' approach, rather than a hierarchical structure with individual 'person' identifiers and passwords, since we believed that passwords are at best minor deterrents. This decision was also based on the fundamental fact that keyboards log in.

We chose to limit our efforts to enhancements that could be appended to or otherwise associated with the LOGIN program.

We designed a 'public key' system; that is, one whose details can be published without compromise. Indeed the details should only serve to frustrate and annoy the talented enemy.

Our system also relies on correct application of a number of RSTS supported feature patches that limit access to certain entities and the entire existing RSTS protection code and privilege/non-privilege dichotomy.

Other design requirements included real-time alarms that would attract attention to all violations and secure methods of disabling lines that are probed during periods of unattended operation. Clear logging of the full text of infraction attempts including all passwords submitted was specified.

Since such a system requires a file that stores keyboard-specific information, the editor for that file must be secured from even the unauthorized privileged user, and laced with checks and alarms to detect its unauthorized use. A full editor command list is in Appendix A.

As we studied the need for system surveillance, it became apparent that login alone was inadequate in the long run, because if it was subverted or compromised, another independent source of information was in order. Consequently, an old classic, DYNPRI was coded in macro to provide a platform from which the dynamic system could be observed for infractions.

As work progressed, we also rewrote another classic "VT5DPY" in macro for VT100. At last, one can observe a system without destroying it! We added a group of UT commands that execute without leaving the program and use the "name" instead of the KB number.

IMPLEMENTATION

The security file was designed initially to contain several sections. The first section contains inversions of the detail contained in the second. These include lists of KB names and lists of KB's with special attributes, and internal security information. The second section contains detail parameters for each keyboard on the system. This includes:

a) Any special attributes (Priv, Alarm, Console, Watch, User-id)

b) Auto-Login information (including Chain and Core Common). Special access-denied message.

c) Priority/Burst, protection codes, @ assignment, 3 user logicals, default RTS.

d) Detail of unlocked P,Pn's with time and date restrictions.

The append code causes LOGIN to perform a number of checks. First, it determines whether or not the terminal is privileged. Should the PRIV attribute be set, LOGIN proceeds normally. That is, normal password access is allowed, even to privileged accounts. All special features such as auto-login, of course, are available. Lacking the PRIV attribute, LOGIN checks the validity of the LOGIN request in the detail section of the security file: Is the project unlocked, the programmer number, the time of day, the day of week, etc. Infractions are labeled and broadcast to a group of keyboards defined as ALARM by the system manager. No hint is given to the user that anything other than "vanilla" LOGIN is in process.

The file editor program is really nothing more than a very long and thorough command parser that updates the security file. It does contain some interesting security measures. It may only be run from a designated "CONSOLE" keyboard. Privilege is necessary but not sufficient. Privileged users attempting to utilize the editor at non-CONSOLE keyboards will cause alarms.

The editor is sensitive to attempts to subvert the file's integrity. Any movement of the file, via PIP or whatever, will cause alarms.

The file itself is secured at its creation by the implantation of hidden validity checks. The file is protected in the UFD by the setting for the protect bit, thus inhibiting renaming and deletion. A special program for creating and destroying the file is in the distribution. This program destroys itself after running and so tends to exist only on the distribution. In order to run this program at all, a special password is required which is only given over the phone to a known licensee.

All code distributed is either Macro or Basic-Plus rendered into Macro via one of the commercial Basic-Plus alternatives. This was a conscious choice since far more is known about Basic-Plus-2 internals and structures.

The LOGIN changes are of necessity, in Basic-Plus. It is the system manager's responsibility to insure that their lifespan on the system is very short. They reveal some of the internal structure, but few of the integrity features.

FUTURES

After a few releases, it became clear that systems with either large numbers of dial-in lines, intelligent port selectors, patch panels or packet-network interfaces would require additional user definition. Allowing all your dial-in accounts on all your dial-in and network lines dilutes security. We then implemented a "user-id" concept. When a line has been flagged as "user-id", Login will start the dialog by requesting a user-id. This ID maps to a specific set of security parameters defined for that user alone; rather than that specific keyboard.

Our wish list for other future developments probably exceeds our current resources:

- alternate banners
- multi-level "Watch" implementation
- time and day limits on PRIV and CONSOLE
- total login rewrite
- project level alarms
- better system manager interface via the DPY program; and, of course, "we have to do it for VAX"

APPENDIX A

COMMAND	USAGE
ACCESS KBn: a/b/h	Set a retries before access-denied b access-denied before disable h = y = hangup if access-denied if dial-up
ACCESS KBn:	Reset to login standard.
AUTO KBn:	Set for no automatic login feature
AUTO KBn:[p,pn]passwd	Set for automatic login to account, the password must be specified here. Prompt for password if omitted.
BURST KBn:<value>	Allows you to change a users login run BURST
CHAIN KBn: <filespec>	Chain to named program (Auto-login must be enabled)
CLEAN	Verify file inversions, rebuilding as required
CLEAR KBn: a1 {,a2,a3...}	Clear attribute status bits for named attributes only
CLEAR KBn: ALL	Clear all attributes
COPY KBn:=KBm:	Copy lock/unlock and logical information only from KBn: to KBm:
CORE KBn: <text>	Load core common text for chain
DEFINE <user-id:>	Add a user-id to list
DENY KBn: <text>	Assign private 'Access denied' message (default is '?Access denied')
DUMP <filespec>	Generate command file mirror image
EXIT	Exit LOCK-11 editor
LIST ALL	Full list, all details, all keyboards
LIST a1 {,a2,a3...}	List all keyboards set for named attributes. Attributes are: PRIV, ALARM, CONSOLE, WATCH, USERID
LIST KBn:	List all details for KBn:
LIST NAME	List all keyboards' names
LIST SYSTEM	List system parameters
LIST <option>/O:<filespec>	Send output to device or file
LOAD <filespec>	Load file using command file
LOCK KBn:[p,pn]	Deny KBn: access to [p,pn]
LOCK KBn:[p,*]	Deny KBn: access to project p
LOCK KBn:[*,*]	Deny KBn: access anywhere
LOGICAL KBn:<prot>	Assign default output protection code
LOGICAL KBn:[p,pn]	Assign @ with PPN, deassign @ if PPN is omitted
LOGICAL KBn:dev:[p,pn]name	Logical assignment, deassignment if no device: and no PPN given. Three user logicals allowed per KB:
NAME KBn:<name>	Assign keyboard name
NAME KBn:	Default keyboard name
PRIORITY KBn:<value>	Allows you to change a users login job PRIORITY
REMOVE <user-id:>	Remove user-id from list

RTS KBn:<name>	Assign default RTS for this KB
RTS KBn:	Clear to system default
SET KBn: a1 [,a2,a3...]	Set attribute status bits for named attributes only
SYSTEM LAST:n	Assign last KB: on the system
SYSTEM PASS:<password>	Assigns the LOCK-11 MOVE password
UNLOCK KBn:[p,pn]	Allow KBn: access to [p,pn]
UNLOCK KBn:[p,*]	Allow KBn: access to all project p
UNLOCK KBn:[p,pn]/FROM=hh.mm, TO=hh.mm	Allow KBn: access to [p,pn] during the time window specified
UNLOCK KBn:[p,pn]/DAYS=day1,day2,day3-day4	Allow KBn: access to [p,pn] for specified days Note that a single UNLOCK command can use both /FROM and /DAYS
ZERO KBn:	Disable KBn: All settings are returned to their default values
ZERO ALL	Initialize the entire file

NOTE: for user-id specification, USER-ID: may be substituted for KB: in any command. The only allowable attribute for user-id is console. A KB: that is marked "user-id" cannot be PRIV or CONSOLE.

EVER MAKE A MISTEAK

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

Once upon a time an attempt was made to lower the priority of a detached compute bound job on Erskine's PDP 11/34 RSTS system. Unfortunatly a mistake was made and this job's priority was set to a value ABOVE all other jobs. To recover from this error, the system could be crashed and restarted or all users could wait until the detached compute bound job was done. Since the compute bound job was going to lock up the system for several additional hours, the system was crashed. Not only did this waste the run-time the compute bound job had already received but it also made many other users unhappy! At least the system was alive again after a few minutes of disk cleaning and INITing.

There's a better way!

Dr. James B. Wilkinson of the Erskine Mathematics Department has provided a much better method of recovery, should I ever repeat my error! This method uses the 11/34's KY 11-LB front panel to halt the system (making sure the system was halted in user mode), to deposit an odd address in the program counter, and to let the system continue. This causes some job to bomb out with a "?Program lost-Sorry" fatal error. Since the high priority compute bound job is most likely the job in question, it gets killed and the system is back to normal for other users.

Both GOTO's in the following procedure should not be required since there is a high probability of getting what is desired the first time.

```
START:  CNTRL/HLT
        CLR
        777776          {address status word}

        LAD
        EXAM            {display status word}

        IF NOT (DISPLAY = 17xxxx OR DISPLAY = 14xxxx)
          THEN
                CNTRL/CONT
                GOTO START
USER.MODE:
        CLR
        777707
        LAD             {address program counter}

        1
        DEP             {deposit odd address in program counter}

        CNTRL/CONT

END:    GOTO START IF problem job is not killed
```

EMPLOYERS:

STOP

PAYING EMPLOYMENT AGENCY FEES!

WHY pay an employment agency a fee of up to 30% of an annual salary to find a qualified computer professional, when you can save by doing it yourself, through our network?

WE'RE CPN, the Computerized Personnel Network. We are **not** an employment agency, so we do not charge commissions. We're a nationwide employment data base, designed specifically for computer personnel.

YOU DECIDE what the specifications are for the position you need filled: job skills, geographical location, salary range, academic background, experience (hardware, software, operating systems and language requirements), etc. Simply enter this information on your terminal, and access our data base through TYMNET or WATS. If you do not have a terminal, we'll provide you with a questionnaire to fill out, and upon its receipt, we'll access our data base for you.

NO EMPLOYMENT AGENCY COMMISSIONS

NO SUBSCRIPTION FEES, OR COMMUNICATIONS COSTS.

WE CHARGE $60 an hour, or approximately $2.50 per resume.

For more information, contact
Douglas Weidner,
Director
1-800/354-9206
In Kentucky call collect, 606-223-4444

Computerized
Personnel Network
Corporate Center
801 Corporate Drive
P.O. Box 4097
Lexington, KY 40544

COMPUTERIZED PERSONNEL NETWORK

CIRCLE 131 ON READER CARD

COMPUTER PERSONNEL:

expose yourself!

for $5

WHETHER you are actively seeking a career move, or merely want to expose your skills, experience and credentials to hiring employers . . . get on the network.

WE'RE CPN, the Computerized Personnel Network—a nationwide employment data base for computer professionals. As an applicant your skills, job experience, academic background and other key information will be entered and stored in our computerized personnel network. The fee is only $5. This on-line data base, but not your personal identity, is accessible by client companies nationwide.

WE ARE NOT an employment agency. We do not charge commissions. Hiring companies can search our data base for potential employees at a nominal cost of $60 per hour (approximately $2.50 per resume), rather than pay employment agency fees of up to 30% of an annual salary.

Call 1-800/354-9206. In Kentucky call collect, 606-223-4444.

THE CALL is toll free. We will send you a full explanation of how the system works and what it can do for you.

Or write:
Computerized Personnel Network
Corporate Center
801 Corporate Drive
P.O. Box 4097
Lexington, KY 40544

COMPLETE CONFIDENTIALITY

CPN

COMPUTERIZED PERSONNEL NETWORK

CIRCLE 131 ON READER CARD

CCLMAN-CCL Manager for RSTS/E

. . . continued from page 18

```
10      !               PROGRAM    :  CCL.B2S                  &
        !                AUTHOR    :  PJH  (OLFBP)             &
        !                EDIT      :  V07                      &
        !                REVISION  :  07                       &
        !                                                      &
        ! ==================================================== &
50      DIM M$(30%)                         !FIP STUFF         &

100     ON ERROR GOTO  19000                !SET UP COMMON ERROR &

110     GOSUB 20000                         !GET COMMON STUFF    &
\       PRINT ID.STG$;"  V";VERSION$;"-";VER.EDT$;" ",INSTAL$  &
\       OPEN "_KB:CCLMAN.CMD" FOR INPUT AS FILE 1%             &

150     GOTO 32700 IF (CCL.ENTRY%<>0%) AND (INDIRECT%=0%)      &
\       PRINT "#"; UNLESS INDIRECT%                            &
\       REMOVE%=0%                          !CLEAR CALL FLAGS  &
\       LINPUT #1%, CMD$                                       &
\       PRINT "@";CMD$ IF INDIRECT%         !ECHO IF FROM FILE &
\       GOTO 150 IF CMD$=""                 !NOTHING ENTERED   &

160     CMD.TYPE$=LEFT(CMD$,1%)                                &
\       I%=INSTR(1%,"ARCHL@E",CMD.TYPE$)                       &
\       GOTO 200 IF I%                      !VALID CMD RECIEVED &
\       PRINT FNERROR$("Invalid command, type HELP")           &
\       GOTO 150                                               &

200     OPEN OPEN.SPEC$+"CCL.DAT" AS FILE #2%,                 &
                ORGANIZATION INDEXED FIXED,                    &
                ACCESS MODIFY,                                 &
                ALLOW  MODIFY,                                 &
                MAP CCLMAP,                                    &
                FILESIZE 500%,                                 &
                PRIMARY KEY CCL.KEY$     UNLESS OPEN.CCL%      &
\       OPEN.CCL% = -1%                                        &
\       MAP (CCLMAP) CCL.KEY$=9%, CCL.APPEND$=9%, CCL.CMD$=50% &
                                                               &
\       I2%=INSTR(1%,CMD$," ")                                 &
\       GOTO 210 IF (I2%<>0%) OR (I%>2%)        !REAL CMD THERE &
\       PRINT FNERROR$("Argument required for function")       &
\       GOTO 150                                               &

210     CMD.VAL$=CVT$$(RIGHT(CMD$,I2%+1%),2%) IF I2% !NO SPACES &
                                                               &
\       ON I% GOSUB 1000,2000,3000,4000,5000,6000,7000         &
\       GOTO 150                                               &
1000    !ADD CCL SECTION                                       &
        GOSUB 20300                         !BREAK CMD TO PARTS &
\       RETURN IF ERROR%                                       &

1010    CCL.KEY$=WS.KEY$+SPACE$(9%)                            &
\       CCL.APPEND$=WS.APP$+SPACE$(9%)                         &
\       CCL.CMD$=WS.CMD$+SPACE$(50%)                           &

1020    PUT #2%                             !TRY TO PUT RECORD &
\       RETURN                                                 &

1030    !DUP KEY ERROR                                         &
        PRINT FNERROR$("CCL "+CCL.KEY$+" exists- Cannot ADD")  &
\       RETURN                                                 &
2000    !REMOVE CCL SECTION                                    &
        REMOVE%=-1%                         !SETUP CALL        &
\       GOSUB 20300                         !BREAK CMD TO PARTS &
\       RETURN IF ERROR%                                       &
\       CCL.KEY$=WS.KEY$+SPACE$(9%)                            &

2010    GET #2%, KEY#0% EQ CCL.KEY$                            &

2020    DELETE #2%                                             &
\       RETURN                                                 &

2040    PRINT FNERROR$("No such CCL found")                    &
\       RETURN                                                 &

3000    !CLEAR CCL TABLE, REQUIRE CONFIRMATION                 &
        PRINT "Really clear CCL table <NO> ? ";                &
\       INPUT #1%, ANS$                                        &
\       RETURN IF ANS$<>"YES"                                  &
\       OPEN OPEN.SPEC$+"CCL.DAT" FOR OUTPUT AS FILE #2%,      &
                ORGANIZATION INDEXED FIXED,                    &
                ACCESS MODIFY,                                 &
                ALLOW  MODIFY,                                 &
                MAP CCLMAP,                                    &
                FILESIZE 500%,                                 &
                PRIMARY KEY CCL.KEY$                           &
\       CLOSE #2%                                              &
\       OPEN.CCL%=0%                                           &
\       PRINT "Table is cleared"                               &
\       RETURN                                                 &

4000    !HELP MESSAGE                                          &
        PRINT "Commands to ";ID.STG$;" V";VERSION$;"-";VER.EDT$; &
                           " follow:"                          &
\       PRINT                                                  &
\       PRINT "A)dd    - Add a new CCL to the table"           &
\       PRINT "R)emove - Remove a CCL from the table"          &
\       PRINT "C)lear  - Clear the CCL table - BEWARE!!"       &
\       PRINT "L)ist   - List CCLs in this table"              &
\       PRINT "E)xit   - Exit from program"                    &
\       PRINT "@       - Indirect commands follow from FILE"   &
\       PRINT                                                  &
\       RETURN                                                 &

5000    !LIST CCLS                                             &
        CCL.KEY$="          "               !START FILE        &
\       FIND #2%, KEY#0% GE CCL.KEY$                           &

5010    GET #2%                             !SEQUENTIAL GET    &

5020    WS.CCL$=CVT$$(CCL.KEY$,136%)                           &
\       WS.APPEND$=CVT$$(CCL.APPEND$,136%)                     &
\       WS.CMD$=CVT$$(CCL.CMD$,136%)                           &
\       PRINT WS.CCL$;"-";WS.APPEND$;TAB(11%);"=";WS.CMD$      &
\       GOTO 5010                                              &

5030    !EOF ON CCLS                                           &
        PRINT                                                  &
\       RETURN                                                 &

6000    !@ INDIRECT COMMAND PROCESSOR                          &
        GOTO 6100 IF INDIRECT%                                 &
\       CMD.FILE$=CVT$$(RIGHT(CMD$,2%),136%) !GET FILE NAME    &
\       OPEN CMD.FILE$ FOR INPUT AS FILE 1%                    &
\       INDIRECT%=-1%                                          &
\       RETURN                                                 &

6100    PRINT FNERROR$("Too many open indirect files - IGNORED ") &
\       RETURN                                                 &

6200    !INDIRECT COMMAND FILE CLOSE SECTION                   &
        OPEN "_KB:CCLMAN.CMD" FOR INPUT AS FILE 1%             &
\       INDIRECT%=0%                                           &
\       GOTO 150                                               &

6300    !INDIRECT FILE OPEN ERROR                              &
        PRINT FNERROR$("Open error on indirect file")          &
\       GOTO 6200                                              &
                                                               &

7000    !EXIT                                                  &
        CLOSE #1%,#2%                                          &
\       GOTO 32700                                             &

10000   !CCL CALL COMES HERE                                   &
        OPEN OPEN.SPEC$+"CCL.DAT" AS FILE #2%,                 &
                ORGANIZATION INDEXED FIXED,                    &
                ACCESS MODIFY,                                 &
                ALLOW  MODIFY,                                 &
                MAP CCLMAP,                                    &
                FILESIZE 500%,                                 &
                PRIMARY KEY CCL.KEY$                           &
\       OPEN.CCL% = -1%                                        &
\       TERM$=""                                               &
\       CMD.VAL$=""                                            &
\       CCL.ARG$=""                                            &
\       CCL.KEY$=LEFT(CMD$,1%)+SPACE$(8%) !ASSUME NO ARGS      &
\       CCL.MATCH$=CMD$                                        &
\       I%=INSTR(1%,CMD$," ")           !WE HAVE A SPACE TERM? &
\       I2%=INSTR(1%,CMD$,"/")          !..OR MAYBE A SLASH??? &
\       I%=I2% IF ((I%=0%) OR (I2%<I%)) !WHICHEVER WAS FIRST   &
                AND (I2%<>0%)                                  &
\       GOTO 10005 IF I%=0%             !..NONE, SO NO ARGS    &
\       TERM$=MID(CMD$,I%,1%)           !SAVE TERMINATOR       &
\       CCL.ARG$=RIGHT(CMD$,I%+1%)      !GET SPEC CALLED       &
\       CCL.MATCH$=LEFT(CMD$,I%-1%)     !SETUP KEY             &
\       CCL.KEY$=LEFT(CCL.MATCH$,1%)+SPACE$(8%)                &

10005   MATCH.LENGTH%=LEN(CCL.MATCH$)                          &

10010   GET #2%, KEY#0% GE CCL.KEY$     !GET SOMETHING CLOSE   &
\       GOTO 10020                                             &

10015   GET #2%                                                &

10020   CMD.VAL$=CVT$$(CCL.KEY$,2%)+"-"+CVT$$(CCL.APPEND$,2%)+"="+ &
                CVT$$(CCL.CMD$,2%)      !SETUP COMMAND FOUND   &
\       CCL.NAME$=CVT$$(CCL.KEY$,2%)+CVT$$(CCL.APPEND$,2%)     &
\       TMP.KEY$=CVT$$(CCL.KEY$,2%)                            &
\       MIN.LENGTH%=LEN(TMP.KEY$)       !CANT LET USER TYPE LESS THAN THIS &
\       GOTO 10100 IF LEFT(CCL.MATCH$,1%)<>LEFT(CCL.NAME$,1%)  &
\       GOTO 10015 IF MATCH.LENGTH% < MIN.LENGTH%              &
\       GOTO 10015 IF LEFT(CCL.NAME$,MATCH.LENGTH%)<>CCL.MATCH$ &

10025   GOSUB 20300                                            &
\       DUMMY$=SYS(PRIV.OFF$)                                  &
\       DUMMY$=SYS(PRIV.ON$) IF WS.PRIV%                       &
\       CORE$= CCL.NAME$+TERM$+CCL.ARG$                        &
\       DUMMY$=SYS(CHR$(8%)+CORE$)      !PUT CORE COMMON       &

10030   I%=INSTR(1%,WS.FILE$,".???")    !WILDCARD EXT?         &
\       WS.FILE$=LEFT(WS.FILE$,I%-1%) IF I% !YEP, ANY EXT WILL DO &
\ PRINT "CHAINING TO '";WS.FILE$;"' AT LINE ";WS.LINE%         IF DEBUG% &
\ PRINT "CORE COMMON '";CORE$;"', PRIV SETTING = ";WS.PRIV%    IF DEBUG% &
\ SLEEP 3%                                                     IF DEBUG% &
\       CHAIN WS.FILE$ LINE WS.LINE%                           &
\       STOP                                                   &

10090   PRINT FNERROR$("Can't find program to run")            &
\       GOTO 32700                                             &

10100   PRINT FNERROR$("CCL not found")                        &
\       GOTO 32700                                             &

10110   PRINT FNERROR$("Filespec error (LOGICAL not defined?)") &
\       GOTO 32700                                             &

10120   PRINT FNERROR$("?Protection Violation")                 &
\       GOTO 32700                                             &

19000   !ERROR PROCESSING SECTION                              &

19050   RESUME 7000 IF ERL=150 AND INDIRECT%=0%                &
\       RESUME 6200 IF ERL=150 AND INDIRECT%                   &
\       RESUME 6300 IF ERL=6000             !INDIRECT STUFF    &
\       RESUME 20385 IF ERL=20380 AND ERR=52                   &
\       RESUME 20397 IF ERL=20380           !FNAME ERROR       &
\       RESUME 32600 IF ERL=30000           !FORCE PROT VIOL   &

19060   IF ERL=5010 AND ERR=154% THEN                          &
                SLEEP 1%                                       &
\               RESUME 5010                                    &

19070   RESUME 5030 IF (ERL=5010) OR (ERL=5000) !LIST STUFF    &
\       RESUME 1030 IF ERL=1020 AND ERR=134%    !ADD,DUP KEY   &
\       RESUME 2040 IF ERL=2010 AND ERR=155%    !RMV,NO REC    &
\       RESUME 10100 IF ERL=10010 AND ERR=155%  !CCL CALL,NO REC &
\       RESUME 10110 IF ERL=10030 AND (ERR=6% OR ERR=2%) !BAD LOGICAL &
\       RESUME 10100 IF ERL=10015 AND ERR=11%   !EOF            &
\       RESUME 10090 IF ERL=10030 AND ERR=5%    !CHAIN NOT FOUND &
\       RESUME 10120 IF ERL=10030 AND ERR=10%   !protec viol    &
```

. . . continued on page 74

Don't Buy Another VT-100* Until You Compare It To Our "SMALL WONDER"

PRODUCT FEATURES	VT-100	SW10
Printer Port	Opt	STD
Green or White Monitor	Opt	STD
20 MA Current Loop	Opt	STD
Non-Glare Monitor	Opt	STD
Programmable Function Keys	No	STD
Small Footprint for Desk Top Use	No	STD
English Language Set-Up Mode	No	STD
Graphic Character Set	Std	Std
Smooth Scroll	Std	Std
Split Screen	Std	Std
24 Lines/132 Column Characters	OPT	No
Blink & Underline Visual Attributes	OPT	No
Composite Video	STD	No
Double High, Double Wide Characters	STD	No

Standard Unit List Price . . . $1675.

$899.

Available Now!

See Us At
NCC Booth A-419

Call For Your FREE 30 Day Trial Unit Today!

1-800-854-6781

For further information call or send this coupon to:

GTC

General Terminal Corporation
14831 Franklin Avenue, Tustin, CA 92680
(714) 730-0123

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone (_____) _____

(GTC is currently seeking Independent Sales Organizations. Inquiries welcomed.)

*Registered trade mark of Digital Equipment Corporation

CIRCLE 128 ON READER CARD

THE RSTS CRYSTAL BALL — Part 1

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

This is the first in a series of articles on new RSTS/E updates, undocumented features, and bugs. Most of the more active RSTS/E users hungrily await new releases from DEC. Often these users are rather disappointed at what they see, or don't see, in new versions of RSTS. Many people are concerned about the future directions of RSTS. In this column, I hope to present information which will be of interest to all of these users.

All of the material contained in this column is based on short talks with the RSTS developers, peeks at past and present RSTS sources, a solid knowledge of RSTS internals, and partially on the opinion of myself and others not necessarily associated with DEC. The information presented here is believed to be an accurate picture of the directions in which RSTS is heading, however DEC is under no commitment to support their product in the manner in which I describe it. Keeping these facts in mind, I welcome you to a look into the future ...

While I will try to make this column intelligible to as broad a range of RSTS users as possible, I do not wish to rewrite the book on RSTS system concepts. I intend to present information which is fairly technical in nature, and therefore I expect the reader to have a reasonable understanding of RSTS monitor operations, structures, etc. Also, the reader will find familiarity with MACRO-11 and the PDP-11 instruction set useful.

GENERAL

I am sure the questions that most people are asking currently are about the latest RSTS release, version 7.1. What has changed since 7.0? Internally, quite a number of things, although most of these will not affect the average user.

DEC has done next to nothing to solve the problem of RSTS security (or insecurity, as the case may be). DEC is aware of the problem, but it is highly doubtful that they will do anything about it in the near future. Users are going to have to rely on in-house software, or, better, one of the available security packages. Several such packages exist, however you must know what you are buying. Some are nothing more than patches to existing DEC software. Others, if improperly installed (which is EASY to do) will cause far more security holes than they close up. The wise choice would be to go with something which replaces existing DEC software, and is not written in BASIC-PLUS.

INTERNAL SYSTEM STARTUP CHANGES

When the START (or line-feed) option of INIT is executed to startup RSTS, INIT prints its various prompts and informational messages and builds a "jam" table for the monitor. This is a table of information which is to be "jammed into" the monitor once it is loaded into memory. INIT also makes hundreds of checks of the hardware configuration, system default run-time system, swap files, etc. Finally, INIT moves one or more loading routines to various "safe" places and jumps into them to load the RSTS monitor. Once RSTS gains control, it initializes several minor things (such as the maximum job size for the "null" run-time system, which is set to current SWAP MAX) and forces the terminal service to create a job on KB0:. Under version 7.0, the monitor completes its startup by putting the newly created job in a FIP wait, and dispatching to the login code (LIN). LIN notices that the system disk is not mounted, logs the job into the system library account (normally [1,2]), and then goes and dispatches to mount (MNT) in order to mount the system disk. Under 7.1, the monitor puts the job in a FIP wait, but dispatches to an internal FIP function called STA (for START, naturally). This function calls LIN and then MNT to log in the first job and mount the system disk, and also loads and sets up overlay sections of the monitor which are supposed to be resident.

Overall, the startup code for 7.1 is cleaner, however it is much more complex due to the selective overlay loading, and the new FIP buffer pool scheme. It has been suggested that it is theoretically possible to patch the monitor to make modules resident or non-resident after the SIL has been linked. This has not been tested, and depends on whether or not SILUS is doing some calculations for INIT, or if INIT is also doing these calculations. If the latter is true, it is possible that a module residency table in the monitor could be changed at will and, upon re-booting the SIL, change the modules which are memory resident.

One rather interesting note: Try sitting on control/T while bringing up RSTS, just after INIT(.SYS) finishes any final initialization. You will probably be able to catch your RSTS job in a startup wait, i.e. FP(STA).

TERMINAL SERVICE

Several minor changes were made to the terminal service between 7.0 and 7.1, including support for FMS V1.5, two new terminal features (GAG and BREAK), and multiple private delimiters, all of which were fairly trivial to implement. I can't say much for the new terminal "features", the first of which is a fix for a long-standing oversight, and the second which removes a supposed feature which has always been far more annoying than useful.

The addition of multiple private delimiters is something I am quite pleased with, as it opens the door for user-written command completion input routines. Unfortunately, I doubt if DEC will borrow from TOPS-20, as they seem quite convinced that the VAX DCL implementation is the way to go. You can fool most of the people most of the time . . .

Two new terminal .SPEC calls are available under 7.1, one of which is undocumented. The first call implements multiple private delimiters. The other, undocumented, allows access to the FMS support in the monitor.

Currently, not much information is available on the FMS terminal .SPEC call, since the terminal service does only minor validation on the parameters. The FMS phase of the monitor is what does the real work, and I have not yet had the time to take it apart. The .SPEC sub-function code for the FMS support call is 10 octal. It is possible that the call is detailed in the FMS documentation set, but I doubt it. The call takes several parameters, including a buffer address and length in XRLOC and XRBC. The target keyboard is passed in the same manner as it is for the multiple delimiter call (in XRMOD), and the target keyboard must be attached (not on a hung dial-up line) and open in FMS mode. Perhaps I will devote a future column to RSTS FMS.

One rather serious bug in the 7.0 terminal service has been fixed in 7.1 (a mandatory patch was supplied with 7.0). The bug was in the pseudo keyboard driver, and could hang the system when the job on a PK was killed, if it was "job" at the time. The PK driver neglected to set the L3Q bit indicating that "job" should be dumped, and could theoretically have caused a race condition. In practice, however, the system-hanging situation would almost never happen.

MONITOR CALLS

A new EMT was added in 7.1, and it is currently undocumented. The call is .XPEEK, or extended peek (would you believe XBUF peek?), and allows a job to look at whole segments of memory. Note that I said memory — not monitor memory — since .XPEEK takes 22-bit physical addresses. This makes it possible to peek at the XBUF and non-mapped monitor memory, as well as any job images which may be loaded into memory. The format of the .XPEEK call is as follows:

Data passed:

XRB

Octal	Offset Mnemonic		Octal	Offset Mnemonic
1		count of bytes to transfer	0	XRLEN
3		must be zero, as for a .READ	2	XRBC
5		address of user buffer	4	XRLOC
7	XRBLKM	MSB of address	6	
11		LSW of address	10	XRBLK
13			12	
15			14	

Data returned: *

If successful, the block of memory specified will be returned in the user's buffer. The following errors are possible with .XPEEK:

PRVIOL

For starters, .XPEEK is privileged. You cannot use

.XPEEK to look at locked out memeory, and furthermore, you must specify a physical address and count which define an area of memory which is inside the memory limits of your system.

BADCNT

The first three words of the XRB are illegal, for standard reasons (buffer in R/O highseg, in context area, etc., etc., count illegal, XRBC not zero, etc., etc.) or XRLEN was passed as greater than 17600 octal. THIS IS THE MAXIMUM AMOUNT OF MEMORY YOU CAN PEEK AT WITH .XPEEK (8K bytes - 2 MMU units = 8192. - 128. = 8064. (17600 octal) bytes). If you would like to know why this is, you can look at the .XPEEK "source" code at the end of the article.

Obviously, .XPEEK can be used for many things which DEC didn't intend it to be used for, such as a grow-shrink SYSTAT or SYSDPY type of a program, looking at other jobs' context areas.

Another undocumented feature exists in 7.0 and 7.1 in the .TIME directive. In the System Directives manual, .TIME is documented as returning various timing information in the XRB. The word at XRB + 12 is described as "reserved for future use". This word is in fact not random, but contgains some rather useful information, detailed below.

The .TIME directive takes no arguments, so the call is simply ".TIME". The directive returns the following information in the XRB:

XRB

Octal	Offset Mnemonic		Octal	Offset Mnemonic
1		low word of elapsed CPU time	0	
3		elapsed connect time, minutes	2	
5		low word of kilo core ticks	4	
7		elapsed device time, minutes	6	
11		high word of elapsed CPU time	10	
13		job description word	12	
15		high word of kilo core ticks	14	

Data returned:

The data returned is described in the System Directives manual, with the exception of the following:

XRB + 12

This word is currently returned as a bit pattern, with only the high three bits being significant. The bit assignments are:

Octal Value	Bit	Meaning if set
100000	15	The RTS under which the calling job is running is the job's default RTS.
40000	14	The calling job is detached.
20000	13	The console keyboard of the calling job is a pseudo keyboard.

The above information is quite useful, as you can see, and would otherwise require reasonably complex .PEEK sequences (also meaning that the job would have to be privileged). Note that since .TIME is an EMT and its information is returned in the XRB, there is no [supported] way to use .TIME from BASIC-PLUS. Also note that .TIME is the best way to determine if the calling job is detached, since it is a synchronous call and takes next to no time to execute. Of course, if more information on the job is required, or if the information required is on a different job, then UU.SYS should probably be used.

As far as I know, this works fine in 7.0 and 7.1, and there is a good chance that it will be supported as soon as someone at DEC remembers that the code is there.

The .FSS call now parses the /PROTECTION:n switch. Apparently, DEC has something in mind for those brokets . . . TOPS-20 uses brokets for its directory specifications, and VMS will accept them in lieu of brackets. Support for named directories is included in 7.1 (that's right, directories accessible by name instead of PPN), however the code is far from bug-free, and I wouldn't recommend running it on anything but scratch packs. Version 7.2, scheduled for July-August of this year is definitely not going to have named directory supported. There is still some debate as to whether or not a version 7.3 will come out, or if we will see an 8.0 instead. If in fact there are any version 7 releases past 7.2, I doubt if they will have the named directory support either. In any case, look for 7.3 or 8.0 going to field-test around February-March of 1983.

Named directory support may be generated by editing CONFIG.MAC to include a line defining the symbol NAMDIR to a 1, i.e. NAMDIR = 1. The module RESNME must be included in the SIL (in the EMT phase) from RSTS.OBJ, and once the monitor is built, several binary patches must be installed. The patch locations are ..NME0 through ..NME3. The first two locations, ..NME0 and ..NME1 are in RESNME (in EMT), the third, ..NME2 is in the MNT code (in OVR), and the last, ..NME3, is in the NME code (in OVR). I believe that all of the patch locations should be made into NOP's to enable named directories. A system file, NAMES.SYS, must be created in [0,1]. Here is where the fun starts. This file is a hash table for the names, and I am not sure of its format. For now, you can generate a monitor with named directory support . . . perhaps I will have deciphered the code by the next issue.

MONITOR BUGS

Several bugs exist in 7.1 which DEC has not supplied patches for (yet). One of these is quite annoying and definitely needs attention. The UU.TRM call (set terminal characteristics) sets all of the information passed into the target keyboard's DDB; however, it fails to call the terminal service to update this information. This means that certain characteristics, most importantly terminal speed, are not changed until the next time the keyboard's interface interrupts. This causes all sorts of problems, because a program can't be sure WHEN the terminals characteristics have actually changed. On a terminal which is set to 9600 baud, try typing SET SPEED 4800. Chances are that you will get the "Ready" prompt (or whatever) ungarbled AT 9600 BAUD, and that when the prompt finishes printing, the terminal will be set to 4800. This is a problem which must be cleared up. Please join me in bringing this bug to the attention of DEC by sending in your SPR today!

UTILITIES

The RSX librarian utility distributed with 7.1 contains a new feature which is extremely useful. The librarian can now process universal libraries; i.e., a library which can store ANY type of data from ANY type of file. One of the more useful applications of a universal library is to store subroutine sources for a package or program, extracting them only when they need to be updated. I mention this new feature, although it is documented, because it is so useful. DEC didn't go out of its way to announce the inclusion of universal libraries. In fact, the release notes say that the LBR utility was not modified. Universal libraries are fully documented in the Programmer's Utilities manual.

There is one minor problem with universal library support. The RSX directive to get a file's attribute information is not supported in RSTS. In the released version of the librarian, the RSX directive isn't used. Instead, a call to a SYSLIB routine is made, incorrectly, causing any attribute information associated with the input file to be garbled when inserted into the library. The attributes on any file extracted from the library will be meaningless. Furthermore, if the file did not have any attributes to begin with, the librarian will tack on random ones when it is inserted, and any output files will have these random attributes. This is really just a minor annoyance, however it makes storing RMS files and the like impossible without re-writing the attributes by hand after extraction. The following patch, although not terribly elegant, will solve this problem:

```
File to patch? $LBR.TSK
Base address? $INS
Offset address? 1232
Base    Offset    Old       New?
??????  001232    ??????    ? BR!(Q&377)
??????  001234    010005    ? !Z
Offset address? 1300
Base    Offset    Old       New?
??????  001300    161347    ? <LF>          ;No change, verify only
??????  001302    ??????    ? BR!(Q&377)
??????  001304    016767    ? !C            ;Up-arrow C to exit
```

WHAT'S NEW IN 7.2

I think I shall save most of this topic for next time, however be on the lookout for a re-write of the RSTS scheduler. Rumor has it that DEC will supply a dynamic job scheduler to replace the existing one . . . Get ready to chuck your DYNPRI and LIMIT programs!

CONCLUSION

I hope you have enjoyed this installment of the RSTS Crystal Ball. I will try to continue to present information which is interesting and usefull. If you have any questions, gripes, or suggestions, call or write: Michael C. Greenspon, 9832 Vicar Street, Los Angeles, California 90034, (213) 558-0732

. . .continued on page 58

LOOK at the "tear-out" cards in this issue.
There's subscription cards for you or a friend.
There's a FREE gift for you. Bring in new subscribers and collect rewards. See "BOUNTY HUNTERS" card.
There's a READERS INQUIRY card for your convenience.

RSTS MONITOR INTERNALS

By Jude Suszko and Bob Meyer

This is the first column in what we hope will become a regular feature in the RSTS PROFESSIONAL. Essentially, we are 2 crazy people who have spent many an evening heaping various forms of abuse on the monitor, and in the process have learned a fair amount about RSTS internals. We have both, on many occasions, found that our experience has enabled us to offer useful advice to others faced with unusual (not always reproducible) situations sometimes encountered on RSTS systems.

The purpose of this column is to provide a forum for the exchange of information ranging from fixes for mysterious bugs to novel ways of enhancing the functionality of the monitor. The authors welcome input from anyone wishing to pose a specific question, present the solution to a problem, or describe an interesting feature of his system that didn't come on the SYSGEN tape. Suggested fixes or workarounds printed here will probably differ (if only in availability) from those supplied by DEC.

Obviously, we cannot assume responsibily for the results (or lack thereof) of the patches we expect to be publishing, but we WILL guarantee that all patches published here have been installed and tested on our own systems. (Should anyone installing our patches be caught or killed, Dave Mallery will disavow any knowledge of this column).

Since this is the first column, we are overwhelmed by an absence of mail representing potential contributions, so we'll take the liberty of presenting a few skeletons from our own closets. These will include a simple patch that can help alleviate the ever-present small buffer problem, a description of an elusive bug in the monitor's LOGIN routine, and a simple feature patch to the terminal driver.

One day, while sweating out another in a long line of small-buffer crises, we looked at a UT SNAP dump to see where all the little buggers were hiding. Lo and behold, a surprisingly large number were tied up in the terminal output chains (over 150!). It seems that when there is an adequate supply of small buffers (over 25% of the number genned [adequate?]), old RSTS hands them out like jellybeans at a Reagan testimonial. In this case, an earlybird user had logged in while the system was lightly loaded and started printing a large report on his hardcopy terminal. Shortly after starting, the printer exhausted it's paper supply, and promptly sent an XOFF to the system. Since there were few demands on the small buffer pool at that ungodly hour, RSTS allowed over 80 small buffers to be allocated to that KB's output chain. Meanwhile, the user (doing whatever users do when disasters of their own causing are killing the system) was oblivious to the lack of paper. When the civilized users started logging in at 9-10 AM, the system was running with 80+ fewer buffers than usual (Yes, it was a Monday).

This a specific instance of a more general problem; the small buffer quotas assigned to a device at SYSGEN time are rather loosely enforced. This problem was partially compensated for by logic in the terminal driver that prevents additional buffers from being allocated to a terminal that is in a ↑S state. Unfortunately, this does not prevent the situation described above.

The following patch can help remedy this problem:

```
File to patch?
Module name? TER
Base address? CHKFRE
Offset address? 14
Base     Offset    Old       New?
??????   000014    100004    ?  240
??????   000016    005761    ?  ↑Z
Offset address?  ↑Z
Base address?  ↑Z
Module name?  ↑Z
File to patch?  ↑Z
```

The routine being patched (CHKFRE) is called by the terminal driver to decide whether or not to allocate another small buffer to a terminal's output chain. The altered instruction was a branch that follows a test to see if the terminal is in a ↑S state; if not, the branch was taken to code that allocated another buffer (based on availablity). This patch changes the branch to a NOP, so that the buffer is NEVER allocated if the terminal is over quota, regardless of the number of currently available buffers. The effect of this patch is to reduce the impact of terminal output on the small buffer pool.

Some (artificially reconstituted) history is in order here. We believe that back in days of old when core was gold, it was advantageous to allow a program to maximize use of it's residency by allowing as much output as possible to be moved to the buffer pool before the program swapped out, thus allowing another job to swap in and execute while the terminal driver emptied the printing job's output chain. In those days, this philosophy was acceptable, since swapping (not enough real core) was the basic limiting factor on job count.

Ever since the dawn of the 11/70 age, this is no longer true. Today, an 11/70 with a megabyte+ is not unusual; swapping can be virtually eliminated by buying (CHEAP!) core; slow death by small buffer shortage is the disease of the day. Anyway, the patch above is advantageous ONLY on systems which are not swap-bound; it assumes that a job will remain resident during more frequent bursts spent transferring fewer characters to the terminal driver. If your system is swap-bound, this patch will make your problems MUCH worse. On the other hand, large-memory systems will benefit from this patch since a job will usually remain resident even though it is stalled in a TT state.

Further reduction in small buffer usage can be achieved by a minor edit of the file TTDINT.MAC prior to SYSGEN. The following is an excerpt from this file as supplied by DEC:

```
.SBTTLLOCAL EQUATES THAT ARE GLOBALIZED

BFQ.KB==10.;MAX # OUTPUT BUFFERS ALLOWED ONE KB
BFI.KB==8.;MAX # INPUT BUFFERS ALLOWED ONE KB
BFE.KB==8.;MAX # ECHO BUFFERS ALLOWED ONE KB
```

We altered this file to appear as follows:

```
.SBTTLLOCAL EQUATES THAT ARE GLOBALIZED

BFQ.KB==5.;MAX # OUTPUT BUFFERS ALLOWED ONE KB
BFI.KB==4.;MAX # INPUT BUFFERS ALLOWED ONE KB
BFE.KB==4.;MAX # ECHO BUFFERS ALLOWED ONE KB
```

The result of this is to allow fewer buffers to be allocated to terminal service functions. This alteration is only useful in conjunction with the patch described above; REMEMBER — on small memory systems it will further increase swapping.

The following is a description of a rarely-seen event which can crash a RSTS system that has dial-up lines. (This quotes a recently-submitted SPR; however, the SPR may be too late to eliminate this bug from V7.1.)

If a dial-up user is trying to log in to RSTS, and loses carrier or hangs up after entering PPN and password, the following sequence of events can take place:

1) LOGIN SYScall is issued and placed in the FIP queue. Meanwhile, the terminal driver notices loss of carrier and sets up that keyboard's MODCLK word for a five second timeout.

2) The system is busy today, so the LOGIN request cools its heels in the FIP queue for 5 seconds of wall-clock time.

3) System clock ticks, interrupting at level 6. This happens to be the tick that begins a new second, so clock service calls the terminal driver at its once/second entry point. The terminal driver does a scan of the MODCLK table and finds that the keyboard that lost carrier has exhausted its 5-second grace period, so the driver hangs up the phone and calls DETJOB to detach the job. DETJOB alters the job's IOB by replacing all pointers to the lost line's DDB with pointers to KBFDDB, a 'fake' DDB that exists in read-only territory.

4) The login request finally makes it to the head of the FIP queue; the routine LIN is called to verify PPN and Password. Finding them acceptable, LIN begins to alter the monitor tables to promote the job to logged-in status.

This gets as far as LIN + 144, where LIN tries to put the time-of-day assigned into the job's KB: DDB. Since the IOB has been altered to point at a fake DDB in read-only space, a memory management violation crashes the system.

Possible fixes include having LIN check the DDB prior to altering it, or having LOGIN open the keyboard in guarded mode (16), which keeps DETJOB from altering the job's IOB. In any case, this bug took 2½ years to manifest itself once at my site, so I don't consider it a serious problem. People with heavily-loaded systems and lots of dial-up activity might think otherwise.

While we have not received a response from DEC yet, the idea of having LOGIN open the keyboard in guarded mode (mode 16) is certainly available to the typical user. Those of you with significant dial-up activity take note.

On a lighter side, we have found the following terminal driver patch to be surprisingly useful:

```
File to patch?
Module name?  TER
Base address?  TI$EOT
Offset address?  0
```



```
Base    Offset  Old      New?
??????  000000  012746?  116102
??????  000002  004040?  2
??????  000004  046126?  16202
??????  000006  000010?  JOBTBL
??????  000010  001004?  16202
??????  000012  042761?  10
??????  000014  004040?  126227
??????  000016  000010?  31
??????  000020  000441?  1
??????  000022  042761?  1002
??????  000024  004000?  105061
??????  000026  000010?  2
??????  000030  000427?  207
??????  000032  004567?  ↑Z
Offset address?  ↑Z
Base address?  ↑Z
Module name?  ↑Z
File to patch?  ↑Z
```

When installed in one's terminal driver, typing ↑D from any Prived terminal causes the current job to become detached.

We found this especially useful when debugging programs using Echo Control mode, and during BP2 compiles. . .

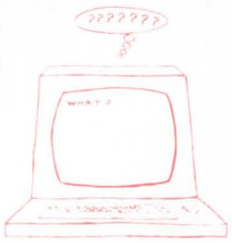
Note: If the terminal is opened in Echo Control mode (mode 8), the ↑D will not take effect until a field is enabled by the program.

Miscellaneous items of potentially useful information. (or 'Did you know that. . .')

Item 1: SLEEP 0%

This call will cause the monitor to re-schedule your job with no fixed delay. It's useful in programming loops that attempt to recover from locked disk blocks (error 19. or error 154. in RMS land). Sleep 0% will give all other jobs run-

DEAR RSTS MAN:



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

DEAR RSTS MAN:

This month's answer is in reference to a letter from Thomas Riesenberg who is Manager of Financial Systems and Programming at Baptist Memorial Hospital in Memphis, Tennessee. Thomas' letter appears on page 82 of this issue.

Dear MM and MT: *The RSTS man thanks you for giving him a chance to expound on at least two subjects: Magnetic tapes and DEC field service.*

Magnetic tapes are touchy devices which can generate lots of loggable RSTS errors. Sometimes these errors are caused by bad or dirty tapes, other times the tape unit or the controller is at fault. I use the following procedure to try to determine where the culprit lies:

1. Clear the ERROR log

2. Put a fresh (new) tape on your tape drive

3. PIP about 5000 blocks to the tape

4. Check the ERROR log for errors.

5. If there are only 1 or two errors then your tape is a good one else . .

6. Repeat the above 5 steps

7. If you still have more than 1 or 2 errors get your drive fixed.

8. Repeat at both 800/1600 BPI

A new tape on a good drive should generate only a couple of errors per 2400 foot reel, more than that and it is not working right.

Now that we have determined that the tape drive is faulty, here is how to handle DEC field service on a recurrent problem that doesn't seem to go away. First, call and log the service problem with that component. If you have a DEC service agreement you have certain rights, you should ask what they are and how they will be implemented. There is a well defined escalation procedure that will move your problem from the branch to the district to the region to national headquarters at specific times; replacement of the unit is also possible after certain fruitless attempts to repair it. You must not accept the faulty unit until it works correctly!

The RSTS man himself recently pushed his field service branch very hard on a TU16 problem. The diagnostics failed to provide and help, but RSTS ERROR logging did show the problem with data errors. Brushing aside their comments that magnetic tapes always cause problems, I insisted they FIX it because as far as I was concerned it was DOWN! They worked 3 days and replaced every major board and mechanical part they could find and in the end I could write a 2400 foot magtape with 1 error in either 800 or 1600 BPI. While they were non-believers in the beginning, they accepted the faith at the end. Push your branch to escalate problems they don't solve, don't give up; learn who to call either in the branch, district, region or Home office.

Develop good feelings that your field service branch is trying as hard as you are to make this thing work — my branch does, they have proved over and over again that they will do everything they can (including replacement of an RP06) to make my system work and work right. Unfortunately, all branches are not equal. If your branch is a little less equal than mine, let me know; MEMPHIS DEC FIELD SERVICE do you hear me? You have a customer who is not happy with you and YOUR tape drives. Please fix them, they can be fixed and made to work right. Just like a branch. ♥

ning at the same priority level a chance to execute, without imposing unnecessary delay on the current job.

Item 2: Null Message Receiver Name

If the receiver name in a RSTS 'declare-receiver' call consists of null bytes, the Send/Receive code will not search the receiver table for duplicate receiver names, nor will it enter a name in the receiver list. Such a receiver can only be sent to via job number; the job has no receiver ID and cannot be sent to by receiver name. This may be useful in situations where a centralized resource manager responds once-and-only-once to any message (requests) for its services, and can send the response message to a particluar job number extracted from the requesting job's pending message block.

We would like to thank you for reading/tolerating this column. We sincerely hope that we can be of help to some folks out there in RSTS-land, as we live there ourselves. Please send any items of interest to:

S & M Enterprises
C/O Jude Suszko/Bob Meyer
9 Lockwood Ave.
Fieldsboro, NJ 08505 ♥

THE DISK INVERSION MAP

By Michael H. Koplitz

A map of the clusters on a device can be developed by reading the disk directory structure that exists on every RSTS/E disk (refer to my article in "**RSTS PRO**", February, 1982 [vol. 4, #1, p.45], "How Do You Read a RSTS/E Disk Structure"). The disk map will indicate whether the clusters are allocated or not allocated. Every allocated cluster will be indicated and the file that it is allocated to the cluster will be printed. Free blocks will be indicated by the term "**FREE**". The MFD will be indicated on the map by the term "**MFD**". The UFDs will be indicated by the term "**UFD**" and the account number of the UFD.

Four programs are involved in creating the disk map. A command file has been produced to ease the burden of running the inversion. The programs are:

Program	Description
MAPUFD.BAS	This program collects the data from the UFDs. The DCS must be changed in this program to match the hardware being used.
MAPUF1.BAS	This program reads the sorted file and expands any entry where the file clustersize is greater than the pack cluster size. There are some statements in this program which must be adjusted to meet the hardware requirements.
MAPUF2.BAS	This program adds the free blocks.
MAPUF3.BAS	This program produces the disk map from the information obtained in the first three programs.

A sort is involved in the procedure. The command file uses SORT-11 to do the sorting. The command file can be altered to use any sort. The command file name is MAPUFD.CMD. The command file was written to use ATPRO. The command file utilizes one argument and that is the device name to map. This can be hard-coded into the command file if so desired. Below is the command file followed by the programs, last is a sample of the report.

```
!*********************************************************************
!*
!* THIS CONTROL FILE PRODUCES THE FULL DISK MAP REPORT.
!*
!* PROCEDURE AND PROGRAMS WRITTEN BY M H KOPLITZ
!*
!* THIS CONTROL FILE NEEDS TWO INPUTS, DEVICE AND ENDING CLUSTER
!*      NUMBER. I.E. @ MAPUFD;DEVICE
!*
!*********************************************************************

RUN #MAPUFD
@@1
RUN $SORT
MAPUFD.SRT/FO:STREAM:50=MAPUFD.DAT/FO:VAR:50/KE:1.5
```

```
^Z
RUN #MAPUF1
RUN #MAPUF2
@@1
20875
RUN #MAPUF3
@@1
QU LP:=MAPUFD.RPT
```

```
\!*****************************************************************
\!*
\!* THIS PROGRAM WAS WRITTEN BY M H KOPLITZ
\!*
\!* THIS PROGRAM WILL GATHER THE UFD DATA NEEDED FOR A FULL DISK
\!*     REPORT.
\!*
\!*****************************************************************

010     DIM #1%,MFD%(3583%,7%)
\       DIM #2%,UFD%(3583%,7%)
\       X$ = SYS(CHR$(6%)+CHR$(-7%))
\       DCS% = 8%                               !DIMENSION VIRTUAL ARRAYS
\                                               !SET UP ^C TRAP
\!*************************************************************
\!*
\!* NOTE THAT THE DCS% MUST BE SET UP PRIOR TO RUNNING THE
\!*     PROGRAM. THE DCS% HERE IS FOR AN RP06.
\!*     DCS% IS THE DEVICE CLUSTER SIZE AND IS HARDWARE DEPENDENT.
\!*
\!* IF ONE HAS SEVERAL DIFFERENT DISK DRIVES THE DCS% VALUE COULD
\!*     INPUTED AS PART OF THE JOB STREAM.
\!*
\!*************************************************************

020     ON ERROR GOTO 32000                     !ERROR FLAGGING SET UP

030     PRINT "MAPUFD   V2.1        Full disk map"
\       PRINT                                   !PRINT BANNER.

035     INPUT "Device";DEVICE$                  !GET DEVICE TO EXAMINE

040     OPEN "MAPUFD.DAT"
                FOR OUTPUT AS FILE #10%
                                                !OPEN OUTPUT.FILE$ FOR OUTPUT

080     OPEN "[1,1]"+DEVICE$
                FOR INPUT AS FILE #1%,
                MODE 8192%                      !OPEN THE MFD READ ONLY

090     PCS% = 16%
\       CS% = 16%
\       FOR X% = 1% TO 7%
\               DISK.CLUSTER
                        = (MFD%(31%,X%) - 1%)
                        * DCS%/PCS%
\               DISK.CLUSTER$
                        = NUM1$(DISK.CLUSTER)
\               DISK.CLUSTER$ =
                  STRING$((5-LEN(DISK.CLUSTER$))
                        ,48%)
                        + DISK.CLUSTER$
\               OUTPUT$ = DISK.CLUSTER$ + ","
                        + "[  0,  0]***MFD***,"
                        + NUM1$(MFD%(31%,0%))
\               OUTPUT$ = OUTPUT$
                        + SPACE$(50%-LEN(OUTPUT$))
\               PRINT #10%,OUTPUT$
                        IF MFD%(31%,X%) <> 0%
\       NEXT X%                                 !WRITE TO OUTPUT FILE THE
\                                               ! MFD CLUSTERS

100     LINK% = MFD%(0%,0%)                     !SET-UP FOR SUBROUTINE

105     GOSUB 15000
\       MFD.LINK% = LINK%                       !GET FIRST NAME ENTRY, GIVE
                                                ! VALUE OF ARRAY


001 !*************************************************************
\   !*
\   !* THIS PROGRAM WAS WRITTEN BY M H KOPLITZ
\   !*
\   !*
\   !* THIS PROGRAM WILL READ THE MAPUFD.SRT FILE AND EXPAND
\   !*  ANY CLUSTERS WHERE THE FILE CLUSTERSIZE IS GREATER
\   !*  THAN THE PACK CLUSTER SIZE.
\   !*
\   !*************************************************************

010     PRINT "MAPUF1  V1.0                     ";
                "   Disk map phase 2"
\       PRINT                                   !PRINT BANNER

020     ON ERROR GOTO 32000
\       OPEN "MAPUFD.SRT" FOR INPUT AS FILE #1%
\       OPEN "MAPUFD.PH1" FOR OUTPUT AS FILE #2%        !OPEN FILES

030     INPUT #1%,A,A$,B$,C
\       GOTO 030 IF A > 20875
\       GOTO 030 IF A - INT(A) <> 0%
\       GOTO 030 IF CVT$$(A$,6%) = ""
                AND CVT$$(B$,6%) = ""
\       PRINT #2%,A;",";A$;",";B$               !READ IN AND PRINT OUT

040     GOTO 030 IF (C/16% - 1%) = 0%
\       GOTO 030 IF C > 256%
\       FOR X% = 1% TO (C/16% - 1%)
\               PRINT #2%,A+X%;",";A$;",";B$
\       NEXT X%                                 !PRINT OUT EXTRA
\!*************************************************************
\!*
\!* CHANGE THE 16 IN THIS STATEMENT TO MATCH YOUR PACK
\!*     CLUSTERSIZE.
\!*
\!*************************************************************

050     GOTO 030                                !GET MORE

32000   IF ERL = 030 THEN RESUME 32767

32010   ON ERROR GOTO 0

32767   CLOSE #1%,#2%
\       END                                     !END UP STUFF


001 !*************************************************************
\   !*
\   !* THIS PROGRAM WAS WRITTEN BY M H KOPLITZ
\   !*
\   !* THIS PART OF THE FULL DISK REPORT ADDS FREE CLUSTERS IN.
\   !*
\   !*************************************************************

010     EXTEND
\ -     PRINT "MAPUF2   V2.1        Phase 3, add in FREE clusters"
\       PRINT                                   !PRINT BANNER

015     INPUT "Device";DEVICE$
\       INPUT "Last cluster + 1 on pack";LAST%  !GET SOME INFORMATION

020     ON ERROR GOTO 32000
\       OPEN "MAPUFD.PH1" FOR INPUT AS FILE #1%
\       OPEN "MAPUFD.PH2" FOR OUTPUT AS FILE #2%        !OPEN FILES.

030     INPUT #1%,A,A$,B$                       !INPUT FROM FILE

037     Z = A
\       GOTO 050 IF A = 1 OR A = 2              !FIRST ENTRY

040     FOR X% = 2% TO (A - 1)
\               PRINT #2%,X%;",";"****FREE****"
\       NEXT X%                                 !FIRST PART OF DISK

050     PRINT #2%,A;",";A$;",";B$               !PRINT OUT OLD LINE

060     INPUT #1%,A,A$,B$                       !INPUT FROM FILE

067     FOR X = (Z + 1) TO (A - 1)
\               PRINT #2%,X;",","****FREE****"
\       NEXT X                                  !PRINT FREE BLOCKS

070     Z = A
\       Y = A
\       Y$ = A$
\       Y1$ = B$
\       GOTO 050                                !GET MORE

32000   IF ERL = 060 THEN RESUME 32700

32010   ON ERROR GOTO 0

32700   FOR X% = Y+1% TO LAST%-1%
\       PRINT #2%,X%;",","****FREE****"
\       NEXT X%
\       PRINT #2%,LAST%;",";"****END****"       !PRINT OUT LAST LINE

32767   CLOSE #1%,#2%
\       END


001     EXTEND
\   !*************************************************************
\   !*
\   !* THIS PROGRAM WAS WRITTEN BY M H KOPLITZ
\   !*
\   !* THIS PROGRAM PRODUCES THE DISK MAP.
\   !*
\   !*************************************************************


010     PRINT "MAPUF3   V2.1          Produce the disk Map"
\       PRINT                                   !PRINT BANNER

015     INPUT "Device";DEVICE$                  !INPUT FOR HEADER

020     ON ERROR GOTO 32000
\       OPEN "MAPUFD.PH2" FOR INPUT AS FILE #1%
\       OPEN "MAPUFD.RPT" FOR OUTPUT AS FILE #2%        !OPEN FILES

050     INPUT LINE #1%,Z$
\       C9% = C9% + 1%
\       Z$ = CVT$$(Z$,4%)
\       Z% = INSTR(1%,Z$,",")
\       A = VAL(LEFT(Z$,Z% - 1%))
\       A$ = RIGHT(Z$,Z% + 1%)                  !GET VALUES

060     START = A
\       START.FILE$ = A$
\       LINE.COUNTER% = 65%
\       GOSUB 1000                              !INIT VALUES

100     INPUT LINE #1%,Z$
\       Z$ = CVT$$(Z$,4%)
```

```
\       Z% = INSTR(1%,Z$,",")
\       C9% = C9% + 1% IF INSTR(1%,Z$,"*FREE") = 0%
\       A = VAL(LEFT(Z$,Z% - 1%))
\       A$ = RIGHT(Z$,Z% + 1%)                          !INPUT FROM FILES

110     GOTO 100 IF A$ = START.FILE$                    !SKIP IF FILE NAME
                                                        ! SAME.

120     PRINT #2%,TAB(COUNTER%*32%);
\       PRINT #2%,USING "#####",START;
\       PRINT #2%,"-";
\       PRINT #2%,USING "#####",A - 1;
\       PRINT #2%," ";
\       PRINT #2%,CVT$$(START.FILE$,128%);              !PRINT OUT OLD FILE

130     COUNTER% = COUNTER% + 1%
\       PRINT #2% IF COUNTER% = 4%
\       GOSUB 1000 IF COUNTER% = 4%
\       COUNTER% = 0% IF COUNTER% = 4%                  !ADJUST COUNTER

140     START = A
\       START.FILE$ = A$
\       GOTO 100                                        !GET MORE


1000    LINE.COUNTER% = LINE.COUNTER% + 1%
\       GOTO 1100 IF LINE.COUNTER% < 64%
\       PRINT #2%,CHR$(12%)
\       P1% = P1% + 1%
\       PRINT #2%,DATE$(0%);"    ";TIME$(0%);
            TAB(45%);"* * * FULL DISK MAP * * *";
            "DEVICE: ";DEVICE$;
            TAB(96%);"PAGE ";P1%
\       PRINT #2%
\       PRINT #2%,TAB(X%*32%);"CLUST RANGE FILE NAME";
            FOR X% = 0% TO 3%
\       PRINT #2%
\       PRINT #2%,TAB(X%*32%);"----------- --------------------";
            FOR X% = 0% TO 3%
\       PRINT #2%
\       LINE.COUNTER% = 5%                              !NEW PAGE

1100    RETURN


32000   IF ERL = 100 THEN RESUME 32700
32010   ON ERROR GOTO 0

32700   PRINT #2%,TAB(COUNTER%*32%);
\       PRINT #2%,USING "#####",START;
\       PRINT #2%,"-";
\       PRINT #2%,USING "#####",START;
\       PRINT #2%," ";
\       PRINT #2%,CVT$$(START.FILE$,128%)               !THIS IS THE END
\                                                       ! STATEMENT

32767   CLOSE #1%,#2%
\       PRINT "TOTAL ALLOCATED CLUSTERS ";C9%
\       END                                             !ALL DONE
```

```
                        * * * FULL DISK MAP * * *      DEVICE:  SY:

CLUST RANGE FILE NAME           CLUST RANGE FILE NAME               CLUST RANGE
FILE NAME           CLUST RANGE FILE NAME
----------- -------------------- ----------- ---------------------- -----------
-------------------- ----------- ----------------
0 - 6       [ 0,0]***MFD***     7-  55      ****FREE****           56 - 56
[ 1,1 ]DIBOLR.STB   57 - 58     [ 1,1 ]DIBOLR.TSK
59-  59     [ 1,1]DBRTKB.CMD    60-  60     [ 1,1 ]DBRSRN.ODL      61 -  27805
****FREE****       27806-27806 ****END****
```

```
SY/-S
RSTS V7.0-08 LAWRENCEVILLE status at 01-Apr-82, 04:53 PM Up: 18:28:08
Job    Who     Where   What    Size    State   Run-Time  Pri/RB   RTS
 1     1,2     Det     ERRCPY   5/28K  SR C25     1:49.5    0/6   BASIC
 2     1,2     Det     OPSRUN  16/28K  SL C27     8:56.1   -8/6   BASIC
 3     1,2     Det     QUMRUN  16/28K  SL C28     1:15.5    0/6   BASIC
 4     1,2     Det     SPLIDL  16/28K  SL C26       0.0    -8/6   BASIC
 5     1,2     Det     SPLIDL  16/28K  SL C24       0.1    -8/6   BASIC
 6     1,2     Det     BATIDL  13/28K  SL C32       0.0    -8/6   BASIC
 7    40,3     KB10    NONAME   2/28K  ^C A02      14.8    -8/6   BASIC
 8   105,0     KB22    GRAPH2   5/28K  ^C          41.0    -8/6   BASIC
 9   118,16    KB13    HW123    3/28K  ^C A09       1.4    -8/6   BASIC
10     1,8     KB25    PTCRSE   9/28K  RN        7:10.1    -8/6   BASIC
11   103,55    KB9     NONAME   2/28K  ^C         12.8    -8/6   BASIC
12     1,0     KB0     SYSTAT  11/28K  RN Lck      5.0    -8/6   BASIC
13     1,8     KB11    CORREC   9/28K  ^C A10    1:10.0    -8/6   BASIC
14    30,37    KB7     BILL     3/28K  ^C         19.1    -8/6   BASIC
20    30,5     KB8     BIN2     3/28K  ^C A13     32.8    -8/6   BASIC
23     1,0     KB26    $99.95  27/28K  KB A14   12:09.9    -8/6   TAX

Ready

RUN LISP
Welcome to LISP/11 v2.10
->
12      KB0     $99.95+TAX      KB(OR)  27(28)K+4K      5.7(+2.3) -8
^C

Ready
```

Submitted by John Partridge, Computer Dept., The Lawrenceville School, Lawrenceville, NJ 08648.

COUNTERATTACK ON PAPERWORK

By Dennis Morgan, Florida Power & Light and Bernie Ward, Florida Computer, Inc.

Florida Power & Light Company (FPL) with its 473 work sites does much more than produce power to serve the nearly 2.2 million customers who are making the state one of the fastest growing, most energy consumptive in the nation. Because of complex accounting procedures and stringent government regulations, the company also generates within itself a staggering amount of paperwork and critical records that must be maintained and often retrieved instantaneously.

Officers at the executive level have long recognized how this growing burden of paperwork and records affects productivity. With the realization that 50 percent of the U.S. white collar work force now engaged in some aspect of the information industry, and that the productivity growth rate in the private sector (in 1979) registered a minus four percent, FPL became determined to reduce record handling costs and increase office efficiency. Thus, in 1978, the Corporate Records Department at FPL opened an aggressive counter attack on the paper problem that confronts business and industry at all levels.

After determining the feasibility of a records retention program and the cost benefits of a corporate centralized micrographics facility, FPL turned to Florida Computer, Inc. (FCI), a Miami-based software development firm.

FCI, under the direction of John H. Wright, provided the Florida utility with a software system that interfaces 16mm Reader-Printers and Digital Equipment Corporation's PDP 11/44 for computer assisted retrieval of microfilmed documents.

The FCI software package allows for direct interfacing of FPL's 3M Microfilm Reader, Reader Printers and the VT100, the newest CRT in the DEC line, and thus provides the capacity to produce copies of required documents. The package includes a Data Entry Subsystem (DES), Query language and Report Writer that could be customized and tailored to fit FPL record requirements.

Since the software works with fiche retrieval units as well, requires no hardware modifications, and is compatible with most DEC operating systems, RSTS/E, RSX-11M and VMS as well as 3M and Kodak equipment, the package allowed FPL a great degree of flexibility in achieving the sweeping revisions it envisioned within its records management system.

In an attempt to unify the FPL records systems, Dennis Morgan, Manager of Corporate Records Services, is directing a five-year records plan which includes a Uniform Filing System, Relative Index, Retention Schedule and Vital Records Programs for the entire company.

Rather than trying to "retrofit" the millions of microfilmed company records, Morgan said his initial objective is to establish new records systems for each department that will handle the normal 10-15 percent annual record growth. In some critical areas, however, retrofitting is an ongoing project.

"We have found that often when we go into the various departments roughly 25 to 30 percent of the paper work can literally be thrown away," Morgan said. "It's mainly duplicate copies, information type copies. Another 20-25 percent we can consign to low cost storage areas. Utilities are not only capital-intensive, they are also very labor-intensive in that we have a lot of people who, because of the many regulations, have to handle a lot of paper".

"The FPL objective is to make jobs less labor-intensive by providing information in the fastest way possible," Morgan said. "I foresee the time when a lot of people working with information will have their own terminals at their desks where they can key in and get the data right off the CRT rather than run all over the department looking for it. But in order to reach that point, you have to first establish the

data base, establish retentions, film the records, index them properly and fit them into the system for quick, accurate retrieval. Essentially, we're still in the early stages of that process."

Few organizations are as record-intensive as are utilities, and within that corporate structure no division faces such rigorous records management demands as do those responsible for producing nuclear energy. It has been estimated that with the Nuclear Regulatory Commission (NRC) and Federal Energy Regulatory Commission (FERC) requirements (in addition to those imposed by the state and the company itself), a minimum of 1400 to 1700 different types of records are generated during the design, construction, testing and final operation of a nuclear power plant. This amounts to millions of vital pieces of paper that must be tracked, retained and instantly retrieved for the life of the plant (40 years).

When FPL launched its comprehensive records management restructuring, a nuclear records specialist was placed in charge of that phase of the project using the PDP 11/44 as the basic tool for the computerized indexing of those millions of records. Originally, the PDP 11/44 came on board solely for nuclear applications, but immediately after installation of the system, projects were added which expanded the nuclear applications of the system. "These additional projects included Turkey Point correspondence for NRC letters and such St. Lucie 1 2 projects as Backfit PC/M Tracing, Backfit Value and Line List, Construction, and Exception List to track systems for turnover. Moreover, NRC requirements have generated further projects expansion with Turkey Point and St. Lucie Backfit operations and Steam Generator Replacement and Engineer Drawing Tracking. The system provides for non-nuclear applications as well, particularly for a potential centralized FPL Records Vault."

Future projects will include: Centralized FPL Records Vault, NRC Compliance or Non-Compliance Tracking and others as requirements create.

Using the 16mm rolled film on cartridges with the 3M automatic page search units, filming is proceeding on day-to-day transactions.

The three operating plants have more than two million records filmed and indexed with an annual increase of 25% in 1980. Another million are awaiting microfilming at the plant under construction.

Approximately two rolls of 2,500 frames each are being used weekly at the three operating plants. This allows for the microfilming of about 5,000 pieces of paper per week per plant. "However," Morgan said, "we are just starting to film at the St. Lucie Unit 2 now under construction. We anticipate the volume there to be twice the amount of records of the other plants combined because of all new regulations. Radiographics, x-rays, purchase orders, vendor specifications, welding reports and a seemingly endless variety of other records must be identifiable and retrievable." "For example," Morgan said, "we not only have to keep track of each weld rod used on that equipment. If necessary we have to go all the way back to the vendor on this material. It's imperative to buy only high quality material, and records must be available certifying this material is high quality. We have to have traceability of all this information so in case of an accident we can go back to find its cause — was it a faulty part, faulty material or faulty workmanship. We must maintain an audit trail."

"We often retrieve records we ordinarily think may never be required. When we shut down for refueling or a repair outage, we take advantage of the reactor's inactivity to repair other equipment. The maintenance people will request information on repairs they made on equipment a year or so ago. They may want to know what they did and how they did it and those records must be available to provide them with that information. In essence, the records retrieval system has become a tool of the maintenance department. If we had to do it manually without the micrographics system, it would probably take 30 minutes, that's if we had the warehouse space to put those millions of records and had an excellent manual indexing system. With the computer system, the entire process might take two minutes. No matter how good the filing system is, there are always misfiles and the advantage of the computer assisted retrieval in microfilming is the system integrity."

FPL is just beginning to move into the COM capabilities provided in the Florida Computer, Inc. software as a backup to the on-line computer assisted retrieval. This further enhances retrievability at many of the company's remote construction sites where on-line computer retrievals are sometimes unavailable.

With the growth of records increasing at an awesome rate, the need for effective records management systems and effectively applied technology is a mandatory requirement for improving productivity in the utility industry. At FPL, the Corporate Records Department is commited to this challenge.

NETWORKING AND THE PDP-11

By Michael H. Koplitz

Networking can be seen in our daily lives. When the telephone is used a vast network of computers is being accessed to connect the telephone call. PDP-11s can be used to create networks, even one as large as the telephone company's. Before a network can be created its designer must be familiar with the concepts of networking. This article will discuss basic network types, message switching, message routing and the hardware/software components needed for a successful network.

BASIC NETWORK TYPES

There are seven basic network configurations, which are listed below. Naturally endless combinations can be devised for the final network. Network configurations are generally devised for economy and need. Networks with the most connections between nodes insure that if a node goes down the network can still function (messages are routed via a different node), but this can be very expensive. Therefore the designer of the network must be aware of the purpose of the network and the budget involved when picking a network type.

1. **Point-to-point** — (figure A) the communication channel is used for only one I/ O device. The I/O device can be a terminal, disk or another processor. The host is connected at one end of the channel and one device is at the other end of the channel. This is the simplest type of network.
2. **Multipoint** — (figure B) a parity line structure in which several I/O devices share the same line. The host is usually designated as the control station. Therefore the host controls the communication channel. The control station uses polling to communicate with the devices on the line. Polling is when the host "invites" the tributary station to send messages at a given time.
3. **Centralized or Star** — (figure C) all of the I/O devices in the network communicate with a central point (the host) that has supervisory control over the network. Users can communicate with each other but only after the supervisor processor has given permission for the communication. Communication is outward from or inward toward the host. If communication becomes necessary between the remote I/O devices, the host acts as a central message switching station to pass the communication between the two points.
4. **Hierarchical or Tree Structure** — (figure D) a heirarchy of computers is used to control and synchronize process and report on the process status. This structure is used in real time applications where sensor based systems are used to monitor and record events on some equipment.
5. **Loop or Ring Structure** — (figure E) the remote stations do not communicate with the host processor individually, instead data is transmitted in a loop around the stations. This structure is economical when several remote stations and host processors are located near each other. It becomes expensive when the equipment is far apart due to Ma Bell telephone lines.
6. **Distributed or Multistar** — (figure F) this configuration consists of several supervisory and/or exchange points. Each point has its own set of users and a means for direct communication between the central points.
7. **Fully Distributed** — (figure G) every node in this system is connected to several neighboring nodes. The additional transmission paths provided by this type of structure improve the overall performance of the network because if one node goes down the entire network need not go down.

MESSAGE SWITCHING AND ROUTING

Message switching and routing involves the method in which a message is sent through the network and how it may be routed to its destination. Computers are generally used to route messages. Telephone companies use PBX and PABX exchanges to route telephone calls (which are actually messages on the network!).

1. **Circuit Switching** — a switching center establishes a direct connection from a terminal to a computer or to another computer. The communication channel is not a constant direct line. This switching is done when dial-up lines are used. After the connection is established, the devices can carry 1-way or 2-way communication. When the communication is finished the switching centers disconnect the circuit.
2. **Message Switching** — each message is sent to the network and is routed to its destination. The message may take different routes to get to its destination. The connections the message may take are established channels.
3. **Packet Switching** — long messages are divided into fixed length segments called packets. The packets of a message may take different routes to get to their final destination.

NETWORK BUILDING BLOCKS

There exists specialized hardware to enable computers to communicate. The major reason for the hardware is that computer networks will generally be using telephone lines as the paths for their messages. Telephone lines use analog signals. Analog signals look like sine waves. Computers use digital signals. Digital signals look like square boxes. Therefore there must be some hardware which will translate the digital signals into analog signals and from analog signals to digital signals.

HARDWARE COMPONENTS

Communication Channels — the paths which are used for transmitting signals. These channels are generally phone lines and are obtained from the common carriers.

a. Narrowband — the communications are transmitted at rates of up to 300 bits per second.
b. Voiceband — communication channel has an effective bandwidth of about 3000 HZ, transmissions can be up to 9,600 bits/second.

INFORMATION PRODUCTS SYSTEMS

High Performance PDP 11/70 Tape Subsystem

Improve PDP 11/70 Tape performance by 400%

The IPS 7016 Subsystem provides Tri-density 800/1600/6250 BPI performance without software changes.

Features:

- ◇ No Software changes required
- ◇ Extended Hardware Error Correction
- ◇ 500 IPS Rewind
- ◇ High Data Recovery
- ◇ Fast Backup
- ◇ Less Tape Mounts
- ◇ Fewer Operator Errors
- ◇ Use Less Tape
- ◇ Reduce Archival Storage Area

Information Products Systems, Inc.
The number one supplier of high performance tape subsystems for the energy exploration market.

Disk and tape subsystems available for DEC, P-E, Raytheon and Data General Computers.
Call 1-(800)-231-7972

Information Products Systems, Inc.
6567 Rookin St./Houston, Texas 77074-5073/ Phone (713) 776-0071/TELEX 792413 IPS-HOU.
Regional Offices in California and Pennsylvania.

CIRCLE 127 ON READER CARD

c. Broadband — communication channel providing data-communication rates higher than the voice band, up to several million bits/second.

Modems — a device which changes a digital (binary) signal into an analog signal so that it can be transmitted on the common carrier lines. The modem also converts the analog signal back to a digital signal at the receiver end. Ma Bell calls them "data phone sets".

Acoustic Couplers — this is an alternative to a modem, it converts the digital signal into an audible tone which can then be transmitted over the common carrier lines. Typical rates are 300 and 600 bits/second.

Line Interfaces — this is an interface between an I/O device and the line in which the communications will take place. The interface usually conforms to the ASCII code and discipline, they also meet the electrical and logical requirements of EIA RS232.

SOFTWARE COMPONENTS

If there is hardware to create the network then there naturally is some software that drives and controls the hardware. This software is usually part of the operating system.

Operating System — a software system which is responsible for any or all of the following functions.

1. Control and monitor program execution.
2. Manage system resources, such as memory and I/O devices.
3. Control input/output devices. The I/O drives would be tailored to work with the network line interface.
4. Store and retrieve data.
5. Store and retrieve programs.
6. Prepare programs in one or more programming languages.

PDP-11 NETWORKS

THE LABORATORY ENVIRONMENT

A network can be created in a laboratory for monitoring the different experiments being performed. Several small PDP-11s (i.e., the small box SB-11) can collect data using the MRRT or RSX-11S operating system then transmit the data to a larger PDP-11 host. The host can perform tests on the data and if any odd circumstances exist a signal can be relayed to the small PDP-11 to perform an additional task. Usually that task is to shutdown the test.

THE INDUSTRIAL ENVIRONMENT

The laboratory environment can be expanded where the small PDP-11s actually control the operations of a shop. The larger PDP-11 host can control invoicing, accounting and manufacturing functions. Data for updating the data bases would be supplied by the small PDP-11s (this has been done at PPG Co. where PDP-11s control the cutting of glass; the sizes to be cut are controlled by an IBM which does the invoicing, PDP-15s using laser technology inspect the glass to determine whether it passes QA regulations).

A corporate data network could be set-up with a PDP-11/70 at the corporate level which collects the necessary

data from the corporation's division. The host site would be able to do any necessary calculations from the offsite computers.

FIGURE A. Point to Point

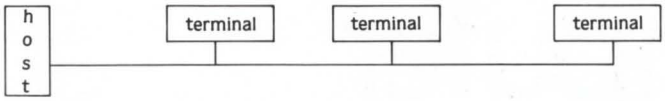


FIGURE B. Multipoint

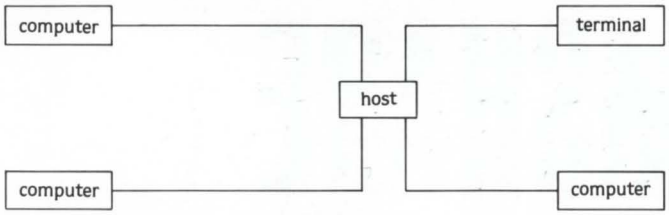


FIGURE C. Star

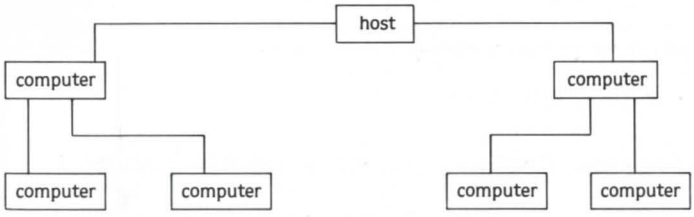


FIGURE D. Tree Structure

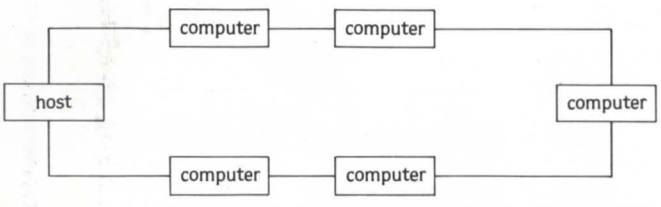


FIGURE E. Loop Structure

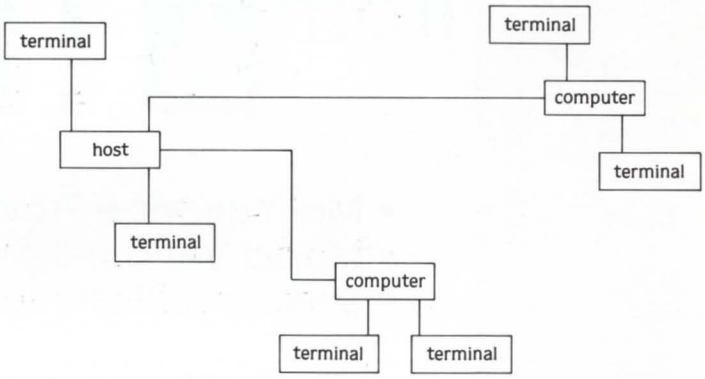


FIGURE F. Distributed

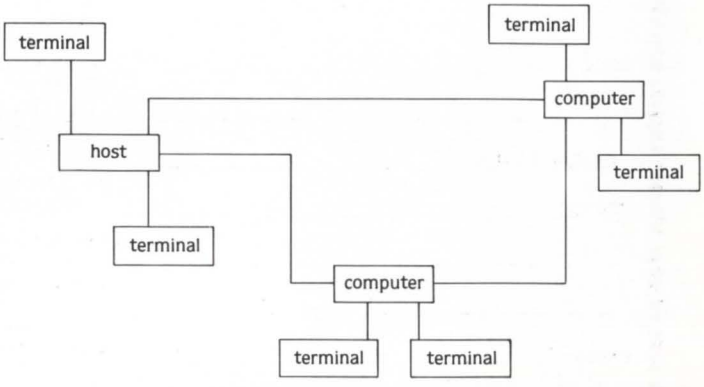


FIGURE G. Fully Distributed

PDP II USERS

WANT AN ALTERNATIVE?
YOU CAN HAVE ONE

with computer maintenance

from TYMSHARE®

- Maintenance Provided on PDPII Systems
- Mixed Vendor Systems Is Our Specialty
- Servicing Most Major U.S. Cities

Don't wait until your system is down again.
For more information write or call now!

Return to: Tymshare Inc. • 3300 Capitol Avenue • Fremont, California 94538 or call 415/794-2528

YOUR NAME _____

YOUR TITLE _____

COMPANY NAME _____

ADDRESS _____

_____

_____

PHONE NO. _____

☐ I WOULD LIKE TO BE CONTACTED BY TYMSHARE TO DISCUSS MY MAINTENANCE NEEDS IN MORE DETAIL.

☐ I WOULD LIKE A PRICE QUOTE ON THE COST TO MAINTAIN MY COMPUTER SYSTEM

TYPE OF MAINTENANCE COVERAGE YOU DESIRE (CHECK ONE)

☐ 8 am to 5 pm Monday through Friday
☐ 8 am to 8 pm Monday through Friday
☐ 24 hour Monday through Friday
☐ 24 hour, 7 days per week
☐ Per Call Only.
☐ Other — Please indicate below:

MY PRESENT CONTRACT EXPIRES _____

EQUIPMENT TYPE	QUANTITY	EQUIPMENT TYPE	QUANTITY

CIRCLE 36 ON READER CARD

USING THE VT100 PRINTER PORT OPTION EFFECTIVELY

By Robert A. Dudley
Meramec Automated Solutions, Inc., St. Louis, MO

The Printer Port Option (VT1XX-AC) for the VT100 Terminal, although scarce at times, can be well worth the nominal expense (and wait). The added flexibility gained with a "sidecar" printer for both development and production use can enhance user throughput and reduce remote communication expenses. The Printer Port option provides several methods to route data to the attached printer, some easy and some more of a hassle.

EASY: Shift Print (Print Full VT100 Screen)
Ctrl/Print (Print one VT100 Line at a time, upon receipt of a Line Terminator)
These modes are selected by user-keyed sequences on the VT100, and are simple to use.

AWKWARD: Printer Controller Mode (Print received DATA directly on Printer)
This mode must be selected by the communications line, and cannot be keyed-in by the user.

If the attached printer is to be used for more than a few lines of printing, such as reports or programs, throughput is of importance. The line-at-a-time mode, although easy to select, is slow in throughput (as much as 50% degradation of line speed) due to concurrent XON/XOFF protocols for both (!) the VT100 and the Printer, since both devices are displaying the data concurrently; each device is independently asserting XOFF's as their associated buffers fill, resulting in an exaggerated stuttering at the printer.

Recognizing this inherent problem, DEC provides the Printer Controller Mode, which passes data directly through the VT100 to the Printer, without displaying the data on the VT100 screen. The only device now supplying XON/XOFF protocol is the printer, and throughput will be the same as if the printer were connected directly to the communications port. A minor disadvantage of this mode is that data input from the terminal is displayed on the printer, rather than the VT100.

The Printer Controller Mode must be selected/deselected by the communications line, meaning that in lieu of a program, the user must do something like a BASIC Immediate Mode command:

PRINT CHR$(155%) + "[5i";

to enable/disable the mode.

(NOTE: Ascii 27 + 128 is more reliable than Ascii 27 for generation of the ESCape Character)

This is particularly inconvenient for those of us who are less than perfect typists, since deselection of the Printer Controller mode requires that what we type is displayed on the printer and not the VT100, where proofreading and error correction can be a supreme inconvenience.

After suffering with this for some time, the author wrote a short BASIC Plus program to accomplish two tasks: 1) Toggle the Printer Port ON or OFF, or 2) Print a Data File through the Printer Port, deselecting the Printer Port after completion of printing or user interruption.

The accompanying Listing gives the BASIC Plus Program. It is intended for use as a CCL PP-RINT = [?,?] PPRINT.BAC;30000. CCL Commands are:

PP[/FF]/ON — Turn the Printer Port Controller Mode ON and issue a Form Feed to Printer if /FF is included.

PP/OFF — Turn the Printer Port Controller Mode OFF.

PP[/FF] FLN — Print the named Filename String through the Printer Controller Mode, then turn the Mode OFF. Issue a Form Feed to Printer prior to printing if /FF is included.

Some Notes:

1) Since ANSI Mode is required for the Printer Port, ANSI Mode is always selected and left on after completion of the CCL Command.
2) If the Program has completed its operation before the user types CTRL/C, the ESCape sequence to deselect the Printer Controller Mode may be cancelled by RSTS emulation of CTRL/O before it can take effect. In this case, use PP/OFF to deselect the mode.
3) The terminal characteristics are temporarily changed to width = 132, with Form Feed Control during the Printing operation. The original terminal characteristics are restored after the operation is complete.
4) I/O error recovery could be vastly improved.

```
1     EXTEND

20    !                 *** PPRINT.BAS *** &
      ! &
      !       MERAMEC AUTOMATED SOLUTIONS, INC.       &
      !                 3637 Scarlet Oak Blvd         &
      !                 St. Louis, MO  63122          &
      !                 ( 314 ) 225 - 3333            &
```

DIGICALC™

The total electronic spread sheet for DEC™ computer systems.

WHY SYSTEMS has developed the "big machine" software package for Executives, Accountants and Professionals who need their numbers fast and accurate.

DIGICALC® is designed exclusively for DEC equipment, with an amazing range of applications and one of the finest built-in automatic training procedures on the market. Prepare everything from budgets to multi-year forecasts, change one value and DIGICALC will re-compute all of your results and give you a hardcopy too. The tabular worksheet display on the wide screen can be as simple or complex as you wish and has the capability to call on numerous mathematical functions. For an amazing display of DIGICALC's ease of operation and virtually limitless capabilities call WHY SYSTEMS. Runs on RSTS/E and VMS.

- ON-LINE HELP AND SELF TEACHING MODE
- TEN KEY NUMERIC DATA ENTRY
- EXTERNAL FILE INTERFACE
- WORKSHEET CONSOLIDATION
- "BOARDROOM QUALITY" REPORTS
- EXTENSIVE MATH FUNCTIONS
  - ALGEBRAIC
  - LOGICAL
  - FUNCTIONAL
  - SCIENTIFIC
  - USER DEFINED FUNCTIONS
- SAVES AND RECALLS WORKSHEETS.

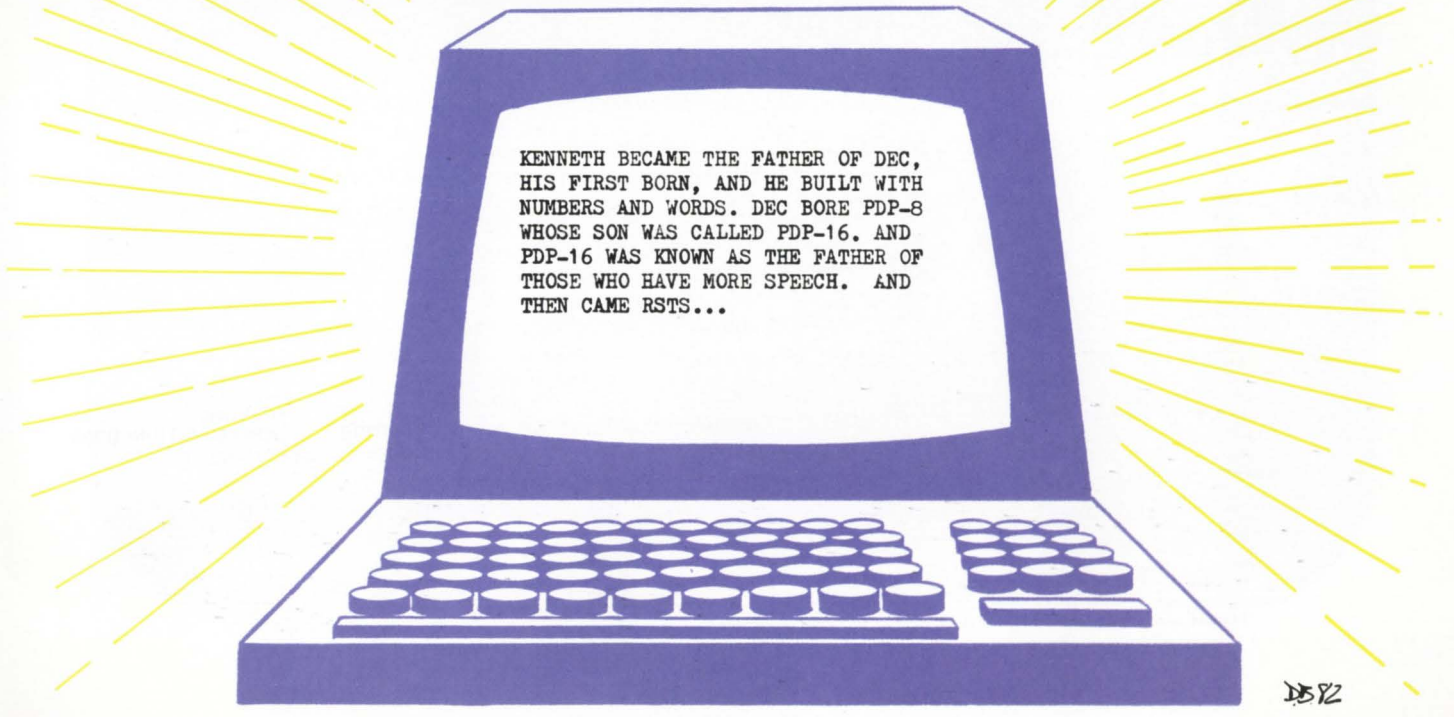
WHY

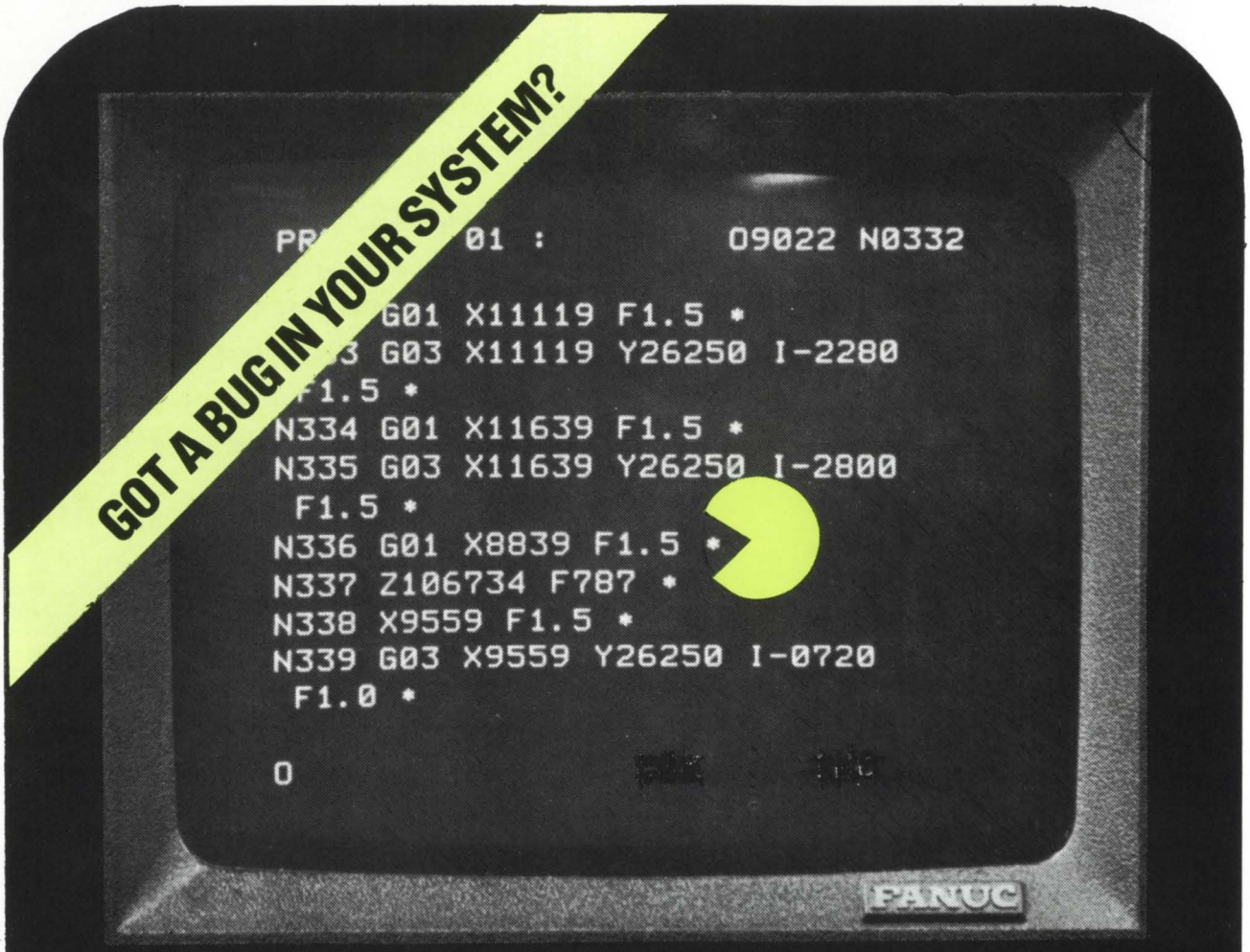
WHY SYSTEMS INCORPORATED
17130 Avondale Way, N.E.
Redmond, WA 98052

CALL TODAY (206) 881-2331 FOR FREE BROCHURE AND DIAL-UP DEMONSTRATION

CIRCLE 72 ON READER CARD

```
        ! &
        !       Program Controller for VT100 Printer Port Option (VT1XX-AC) &
        !       Enter through CCL PP[/FF] FLN, at line 30000, to have a file &
        !       printed through the printer port (/FF will Form Feed printer &
        !       prior to printing file). &
        !       or, PP/ON and PP/OFF to toggle the Printer Port On and Off &
        !               Author: R.A. Dudley &
        !               Date:   DEC 1981 &

30      GOTO 32767 &
                ! Cannot RUN the program &

1000    !       Mainline Processing &
        ! &
        ! &
          OPEN FLN$ FOR INPUT AS FILE 1% &
                \ FIELD #1%, 512% AS LINE.IN$ &
        \ PRINT IF CCPOS( 0% ) &
        \ PRINT "       Printing " + FLN$ + " ( ^C To Stop ) ... " + FNPP.ON$; &
                ! Turn on the printer port &
        \ TRAP.CTRLC$ = CHR$( 6% ) + CHR$( -7% ) &
                ! Save the SYS call to Trap Ctrl/C &
        \ ON ERROR GOTO 19000 &
                ! Standard Error Trap &
        \ DUMMY$ = SYS( TRAP.CTRLC$ ) &
                ! Enable Ctrl/C trap &

1010    UNTIL AN.ERROR$ &
                \ GET #1% &
                \ PRINT LINE.IN$; &
                        ! Get and Put DATA until EOF or User CTRL/C &
        \ NEXT &
                ! This loop is terminated only by ERRor Trap to 19000 &

15000   !       FNPP.ON$, Function to Turn on Printer Port, and &
        !               execute SYS call to set Printer Characteristics &
          DEF* FNPP.ON$ &
        \ DUMMY$ = SYS( PP.SET$ ) &
        \ FNPP.ON$ = PP.ON$ &

15010     FNEND &

15100   !       FNPP.OFF$, Function to Turn off Printer Port, and &
        !               execute SYS call to restore Terminal Characteristics &
          DEF* FNPP.OFF$ &
        \ DUMMY$ = SYS( VT100.SET$ ) &
        \ FNPP.OFF$ = PP.OFF$ &

15110     FNEND &

19000   DUMMY$ = SYS( TRAP.CTRLC$ ) IF ERR = 28% &
                ! Reenable Ctrl/C trap as soon as possible. &
        \ RESUME 19010 &

19010   CLOSE #1% &
                ! Assume we are done, and close-up shop &
        \ PRINT FNPP.OFF$; &
                ! Turn the Printer Port Off &
        \ GOTO 32767 &
                ! That's it Folks &

30000   !       CCL Entry Point: &
        ! &
        !               PP/ON   Turn Printer Controller Mode ON &
        !               PP/OFF  Turn Printer Controller Mode OFF &
        !               PP[/FF] FLN     Print Named File through Printer Port, &
        !                               then turn Mode OFF &
        !                               Form Feed the Printer 1st if /FF &
        ! &
        ! &
          CCL$ = SYS( CHR$( 7% ) ) &
                ! Grab the CCL Command &
        \ ESC.CHR$ = CHR$( 155% ) &
                ! Use Ascii 27+128 as Escape Character &
        \ DUMMY$ = SYS( CHR$( 6% ) + CHR$( 16% ) + CHR$( 0% ) + CHR$( 255% ) &
                                                + STRING$( 28%,0% ) ) &
        \ VT100.SET$ = CHR$( 6% ) + CHR$( 16% ) + RIGHT( DUMMY$,3% ) &
                ! Retrieve the Current Terminal Characteristics which &
                ! have been set by user &
        \ PP.SET$ = LEFT( VT100.SET$,4% ) + CHR$( 133% ) + MID( VT100.SET$,6%,1% ) &
                                          + CHR$( 255% ) + RIGHT( VT100.SET$,8% ) &
                ! Same attributes as terminal, except &
                ! line width of 132, and Forms Control &
        \ PP.ON$  = ESC.CHR$ + "<"    +   ESC.CHR$ + "[5i"  &
                                    +   CHR$( 13% ) + CHR$( 10% ) &
        \ PP.OFF$ =                     ESC.CHR$ + "[4i" &
                ! Strings to Toggle the Printer Port ON/OFF &
                ! PP.ON$ Begins with ESCape Sequence to set VT100 to ANSI Mode, &
                !       and Ends with a New Line to Flush the Printer Buffer &
        \ PP.ON$ = PP.ON$ + CHR$(12%) IF INSTR( 1%,CCL$,"/FF" ) &
                ! Add a Form Feed for Printer if Requested &
        \ IF INSTR( 1%,CCL$,"/ON" ) THEN &
                PRINT FNPP.ON$; &
          ELSE &
                IF INSTR( 1%,CCL$,"/OFF" ) THEN &
                        PRINT FNPP.OFF$; &
                ELSE &
                        FLN$=RIGHT( CCL$,INSTR( 1%,CCL$," " )+1% ) &
                      \ GOTO 1000 &
                              ! If not a Toggler Command, then go &
                              ! process the File to Print &

32767   END
```



"CALL THE RSTS/NCA PERFORMANCE SPECIALISTS."

(213) 286-9510

Stop playing games with your PDP-11

If FIP WAIT and DISK I/O are eating your response time, simply call one of our software specialists to fine-tune your system and optimize your output capabilities

At AST, we understand budget limitations, and we offer full-service RSTS/NCA consulting to the international computer community.

- CUSTOM SYSGENS
- QUARTERLY ENHANCEMENT INSTALLATIONS
- USER TRAINING
- INTERIM SYSTEM MANAGEMENT
- CUSTOM SOFTWARE DEVELOPMENT
- SECURITY PROFILES
- HARDWARE/SOFTWARE PURCHASE ANALYSES

ASSOCIATED SYSTEMS TECHNIQUES, INC.

10528 Lower Azusa Road
Suite 105
El Monte, CA 91731

"Whether it's systems or software, the answer is in the Technique."

CIRCLE 124 ON READER CARD

The first part of Mr. Holmay's article and program, "Logging Into An Account Without LOGIN", appeared in the previous issue of **"RSTS PROFESSIONAL"**, [vol. 4, #2, April 1982, p.8], and was co-authored by Robert Schilmoeller.

JUMP.BAS ENHANCEMENT

By Patrick Holmay
Computation Laboratory, St. John's University
Collegeville, MN 56321

Since the April publication, there have been several enhancements to the above mentioned program. The user now has the ability to do the following:

1) After entering an account, the user has the ability to force input to his/her keyboard by typing a " \ " followed by a CCL or system command. These commands will be executed once the program has logged into the new account. The user may enter more than one CCL or system command separating each by a " \ ". The user must keep in mind that only 255 characters can be forced to the keyboard buffer at one time. If the user chooses to attach to a detached job, any commands entered will not be forced. All commands entered will not be echoed to the user keyboard.

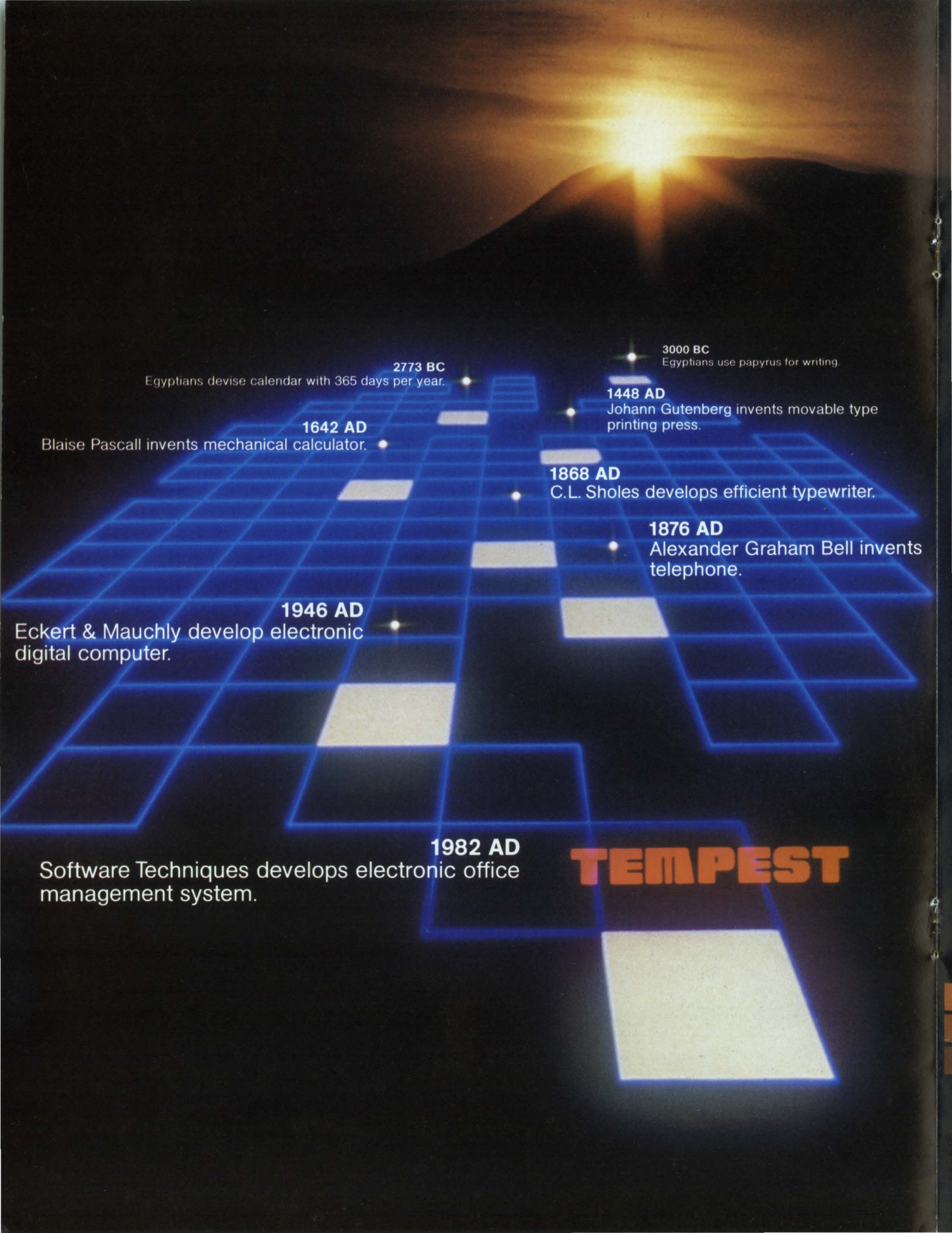
2) If a user has logged into an account with a quota set to one or an account that is over quota, the user has the ability to override the problem of not being able to log out by simply executing the JUMP program. Depending on whether the program is executed via a CCL command or run in normal mode, the user just simply has to type the word 'BYE'. This will eliminate the frustrations of having to JUMP to another account and logging out.

A listing of the various line numbers that need to be added follows.

```
2010    LOGOUT% = INSTR(1%, ACCOUNT$, "BYE")
        \ IF    LOGOUT%
          THEN  GOTO 2030

2012    OPTION% = INSTR(1%, ACCOUNT$, "\")
        \ GOTO 2015     IF      OPTION% = 0%
        \ OPTION$ = CVT$$(RIGHT(ACCOUNT$, OPTION%+1%), 4%)
        \ ACCOUNT$ = LEFT(ACCOUNT$, OPTION%-1%)
                ! CHECK TO SEE IF THE USER HAS REQUESTED
                ! ANY OPTIONS.

2015    COMMA% = INSTR(1%,ACCOUNT$,",")
        \ SLASH% = INSTR(1%,ACCOUNT$,"/")
        \ IF    COMMA%  OR  SLASH%
          THEN  2020
          ELSE  PROJ% = 1%
                \ PROG% = INSTR(1%,WILDCARD$,LEFT(ACCOUNT$,1%))
                \ IF    PROG% > 0%
                  THEN  COMMA% = 1%
                        \ GOTO 2030
                ! DETERMINE PPN SEPARATOR

2017    PRINT "?Can't find file or account"
                \ GOTO 9000
                ! DETERMINE IF USER HAS TYPED IN AN
                ! ACCOUNT # OR A WILDCARD SYMBOL.
```



```
2070    LOGIN$ = SYS(CHR$(6%)+CHR$(4%)+CHR$(0%)+CHR$(0%)+
                CHR$(PROG%)+CHR$(PROJ%)+PASSWORD$)
        \ CHANGE LOGIN$ TO M%
        \ GOTO 8000     IF      RET.PGM$ <> NULL$
        \ GOTO 2080     IF      SLASH%
        \ GOSUB 12000   IF      M%(4%) > 0%
        \ GOSUB 11000
                ! IF USER WANTS TO RETURN TO PROGRAM...GO
                ! ELSE LOGIN USER TO NEW ACCOUNT
                ! CHECK AND SEE IF USER WANTS TO SEE THE NUMBER
                ! OF USERS AND DETACHED JOBS FOR THIS ACCOUNT.

2080    GOSUB 13000     IF      OPTION%
        \ GOTO 9000
                ! DO THEY WANT TO FORCE ANY CCL'S OR COMMANDS
                ! TO THEIR KEYBOARD?

12070   M%(3%) = M%(3%) - 1%
        \ GOSUB 11000
        \ M%(1%), M%(2%) = 6%
        \ M%(3%) = ATT.JOB%
        \ M%(4%) = 0%
        \ M%(5%) = PROG%
        \ M%(6%) = PROJ%
        \ PRINT
        \ PRINT "Attaching to Job";ATT.JOB%
        \ Z$ = SYS(CHR$(6%)+CHR$(-21%)+CHR$(0%))
        \ CHANGE M% TO LOGIN$
        \ Z$ = SYS(LOGIN$)
        \ Z$ = SYS(CHR$(6%)+CHR$(-21%)+CHR$(255%))
        \ RETURN
                ! IF JOB IS DETACHED UNDER THIS ACCOUNT
                ! THEN PRINT THE NUMBER OF USERS LOGGED
                ! IN UNDER THIS ACCOUNT AND ATTEMPT TO
                ! ATTACH TO THE SPECIFIED JOB NUMBER.

13000   !
                ! S T R I P   O F F   O P T I O N S

13010   OPTION% = INSTR(1%, OPTION$, "\")
        \ GOTO 13030    IF      OPTION% = 0%
        \ IF    LEFT(OPTION$, OPTION%-1%) = "$"
        THEN    OPT$ = CHR$(27%)
        ELSE    OPT$ = LEFT(OPTION$, OPTION%-1%) + CHR$(13%)
                ! ALLOW FOR AN ESCAPE TO BE ENTERED

13015   GOSUB 13020
        \ OPTION$ = RIGHT(OPTION$, OPTION%+1%)
        \ GOTO 13010

13020   Z$ = SYS(CHR$(3%)+CHR$(0%)+CHR$(KB.NUMBER%))
        \ Z$ = SYS(CHR$(6%)+CHR$(-4%)+CHR$(KB.NUMBER%)+OPT$)
        \ Z$ = SYS(CHR$(2%)+CHR$(0%)+CHR$(KB.NUMBER%))
        \ RETURN
                ! FORCE COMMANDS TO THE KEYBOARD

13030   OPT$ = OPTION$ + CHR$(13%)
        \GOSUB 13020
        \ RETURN
```


3000 BC
Egyptians use papyrus for writing.
2773 BC
Egyptians devise calendar with 365 days per year.
1448 AD
Johann Gutenberg invents movable type printing press.
1642 AD
Blaise Pascall invents mechanical calculator.
1868 AD
C.L. Sholes develops efficient typewriter.
1876 AD
Alexander Graham Bell invents telephone.
1946 AD
Eckert & Mauchly develop electronic digital computer.
1982 AD
Software Techniques develops electronic office management system.
TEMPEST

MAN'S TOOLS

Since the dawn of civilization, Man has built tools. Tools manage his resources, improve his communications, and help him plan his future. From the stone knife to the laser drill, Man has used tools to expand his knowledge, his capabilities, and his vision. And, with each new horizon, Man has built better tools to meet his changing needs. Tools with more precision, more speed, and more power.

Now there's TEMPEST, the most advanced business tool in 5,000 years. TEMPEST manages your communications with electronic mail. It keeps your messages handy from anywhere in the world and keeps you in touch with everyone in your organization. Instantly.

TEMPEST manages your resources with electronic scheduling. It handles your busy calendar, coordinates meetings with any number of people, and schedules conference facilities and equipment. Automatically.

If you have a PDP-11 computer running RSTS, VMS, or RSX, TEMPEST is the tool you need to cross your next horizon. Please call or write for more information.

Software Techniques Incorporated

Business tools for the work of man.

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

74/76 Northbrook Street
Newbury, Berkshire RG13 1AE
United Kingdom
Phone: 44 [0] 635 30840

CIRCLE 65 ON READER CARD

TIPS & TECHNIQUES

A Column For The Advanced RSTS/E User

Wef Fleischman, Software Techniques, Inc.

COTREES — USING THEM TO ENHANCE YOUR PROGRAMMING PRODUCTIVITY

This column will describe some of the taskbuilder's overlay capabilities and specifically address the use of cotrees. You will be interested in this article if you want to learn how overlaid subroutine libraries can improve your productivity. You will also be interested if you desire to distribute overlaid subroutine packages to other programmers who may have little or no proficiency in overlaying techniques. You may also just want to receive some background on overlay technique and the taskbuilder. With these objectives in mind lets see what cotrees are and how they might be useful to you.

1.0 Introduction - Why are overlays needed?

The RSTS programmer has many tools at his disposal to accomodate programs that are too big to fit into 32KW. Resident libraries (and clusterable libraries some day soon) as well as .PLAS monitor directives can increase your program's effective address space but this is akin to expanding your living room by knocking out the wall to the bedroom: you got the extra space, but it may severly restrict your other activites. Your task might make good use of the memory, but you must always think of all the other things you might need the memory for; e.g., other jobs, small buffers or XBUF.

If one accepts the notion that programs should be subdivided into small, modular units that are sequentially executed (as most of us have I trust), benefits are reaped in simplistic design and ease of debugging. Also, at this point overlaying becomes straightforward.

In theory, overlays reduce the total amount of memory that a task requires by reusing the same address space for multiple modules that do not all need to be loaded into memory simultaneously. In general, a subroutine may be overlaid with any other subroutine as long as neither one calls the other. Since most subroutines that we write only call a few common subroutines, the majority are eligible for being overlaid out of memory when unneeded. For instance in figure 1, modules A and B both are loaded in the same memory area of our task. We say that module A is independent of and overlaid with module B.

The RSX-11M taskbuilder was adapted many years ago to RSTS/E for the purpose of making overlaying relatively easy. The taskbuilder simply requires a specification of what modules are to be overlaid with what others. This specification is provided by you in the ODL file. From there, TKB creates autoload vectors and segment descriptors (the overlay database) and includes modules from SYSLIB to cause overlays to be brought in and out of memory at just the right times for your program to execute properly. All these things are done transparently to you and your task.

2.0 Creating subroutine libraries

Since your program is now broken down into modular subroutines you are ready to collect an added bonus: you can take some of the subroutines and use them in other programs (provided they are written with a general form). As you start using them in different assorted tasks, you save the coding effort and debugging time required before your task finally runs as intended. In this way the programmer can go to bed at 2 A.M. instead of the standard 4 A.M. When you find additional bugs in such a common subroutine, you will not just be fixing a single program, but a whole host of programs. Therefore the use of a subroutine library can greatly improve the quality, maintainability of your programs, and at the same time insulate you from the unexpected appearance of some latent bugs.

3.0 Overlay your library routines and put them in a cotree

In the previous section we saw that subroutines and overlays go hand in hand in the RSTS environment. They tend to create efficient program images (by minimizing consumed memory), encourage good modular programming techniques and increase programmer productivity by providing debugged "building blocks" with which to build many programs.

Lets assume then that you are writing a program consisting of several overlaid program phases, and all phases make use of several handy subroutines: SUB1, SUB2 and SUB3. Since you have found these subroutines useful in previous programs you have written, you have compiled them and kept them available for future programs. Since each major phase (A, B, and C) of your current program will require each of these subroutines, you build them all into the root segment of the task as shown in figure 1. With this overlay structure, these subroutines each occupy separate dedicated areas of memory in the root.

If the subroutines are logically independent of one another (that is to say they don't "CALL" one another) they should be eligible candidates for being overlaid to leave more memory for the rest of the task. This would be possible by building the overlay structure shown in figure 2, specifying each subroutine as an overlaid subroutine in each phase. This method yields the desired memory savings, but has several bad effects: 1) the ODL file has become needlessly complicated; 2) the resultant TSK file has become much larger due to multiple storage of SUB1, SUB2 and SUB3.

A better method of overlaying these subroutines is available and as you have already probably guessed, this method utilizes cotrees. The cotree is simply a supplemental (multiple) overlay structure in your task image. It resides in an independent memory area and may load overlays inde-

pendently of the task's main overlay tree. This means that any module in the main tree (or other cotrees) may call the subroutines located in the cotree with minimum regard for their location, much as if they were all located in the root (as in figure 1, but with the memory savings demonstrated by figure 2).

4.0 Efficient expansion of your user library

A significant payback from the cotree library approach arises as your subroutine library increases in size. Normally a large library of useful routines called into the root becomes impractical to use because too much memory is consumed if you like to call a large number of your library routines. A programmer is likely to limit his use of the library modules because he knows that each additional subroutine he uses diminishes the memory available for the remainder of the task. This contradicts the reason for constructing the library in the first place.

This is why you should provide an ODL file similar to the one shown in figure 3 (USRLIB.ODL), so that future programs can be easily built using all of the commonly used library routines, but without allocating excessive amounts of unnecessary memory, instead allocating them to a cotree. Note that the ODL file shown has two different overlay structures, USEROV and USERAL. The programmer simply specifies USEROV in his .ROOT directive for the user library fully overlaid, or USERAL if he is not particularly concerned with memory usage and would rather have the routines non-overlaid. Any number of variations of overlay degree could likewise be offered. The user would never reference BOTH factors however.

In this way the programmer is encouraged to make use of such subroutines at little additional personal cost.

5.0 Are higher-level language subroutines suitable for libraries?

There is generally the feeling that higher-level language subroutines are not suitable for inclusion in user or group libraries unless they are highly efficient in execution time and memory requirement. This is not strictly true. The distinction should be based on the usefulness of the routine and the number of times it is executed.

A service subroutine that is executed tens, hundreds or thousands of times during execution should indeed be as efficient as possible, probably a good candidate for being coded in MACRO-11, optimized to the last bit.

Infrequently called subroutines, however, rarely impact execution time significantly and therefore should not be ruled out as user library candidates on the basis of size or efficiency. Since inefficient (overly large or overly slow) code often results from higher-level language compilers, the erroneous motivation exists to keep such code out of user or group libraries. Consideration should be given instead to the generality and usefulness of any routine to decide whether it should be in a general purpose library. The primary question to ask is "How much work will this routine save me in the future?"

Since inefficient code can be "hidden" in a cotree overlay as illustrated above, any useful routine, even coded in high-level language may be justified for your library.

6.0 Use cotrees to distribute overlaid subroutine packages

For whatever reason, you may someday need to write a subroutine package for inclusion in a number of different programs, either of our own or those of others. If this package includes a number of modular units, they can most likely take advantage of overlay techniques to lessen their impact on the memory requirements of the resulting task images.

Placing the responsibility on other users to correctly overlay routines that you have designed is a highly precipitous proposition. Without a detailed knowledge of the logical flow of your subroutines, another user may flounder fatally and tell you tales of reserved instruction and odd address traps. The answer of course is to provide a suitable ODL file with the subroutine library that builds the overlay structure into a cotree. The use of a cotree leaves the user free to define his own local overlay structure (and other cotrees) with no conflict with yours.

Does this technique sound familiar? It's exactly the technique DEC uses to distribute overlaid RMS11 code in users' tasks.

7.0 Cotree limitations and "Gotcha's"

Some words of warning should be passed along before we part on the dear subject of cotrees.

First of all, I know of at least one bug in the overlay runtime system that corrupts the overlay database and crashes the task only if the task uses cotrees. As with any system features that have not been extensively tested by users for years upon years, bugs are bound to be uncovered. DEC has promised to fix this particular bug for the next RSTS release following V7.1, but you may need to use PMDUMP to understand the full story.

Second of all, some higher-level languages are not compatible with the way the taskbuilder builds cotreed tasks, so consult your language user's guide and release notes on this subject. If you don't find any specific information, try a few experiments.

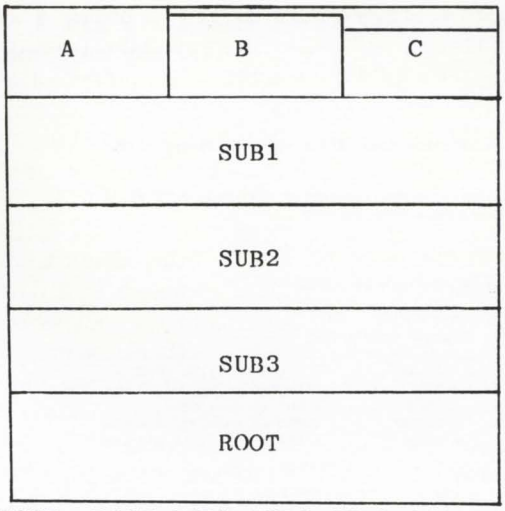
In particular, be aware that the high-level language OTS routines should be allocated explicitly to a permanently loaded area of memory area such as the main root or a cotree root to prevent unintended overlay loading in an adjacent tree. The taskbuilder will identify some of these references by the message "SYMBOL IS AMBIGUOUSLY DEFINED".

Third, read the taskbuilder manual chapter to familiarize yourself with ODL syntax for cotrees, and especially for placement of autoload indicators. Remember in particular that an autoload indicator must be explicitly placed on the root segment of any cotree for it to be loaded into memory.

8.0 Conclusion

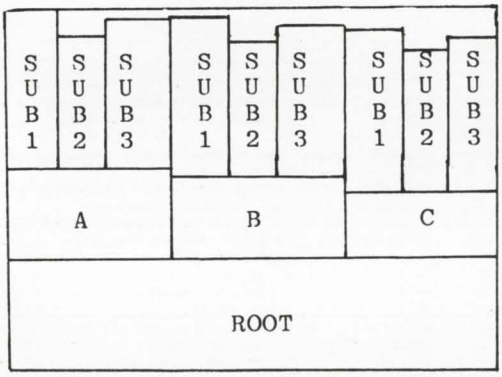
The use of cotrees promotes and encourages good programming habits and is not limited to MACRO programmers or wizards. They remove some of the inconvenient impediments to the use of subroutine libraries and can therefore significantly improve programmer productivity both when generating code and in debugging.

Don't let the taskbuilder's (often) voluminous warning messages put you off. Usually they are very useful advisory messages that aid you in building the overlay structure you want. Until next time, good luck!



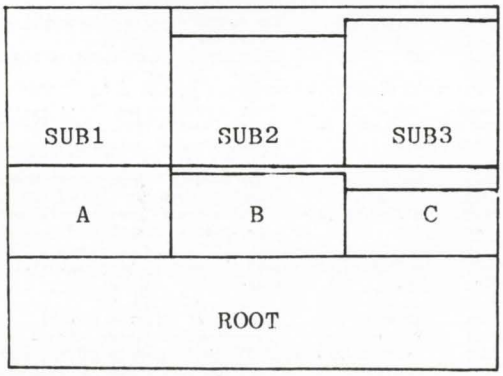

```
.ROOT   ROOT-SUB1-SUB2-SUB3-*(A,B,C)
.END
```

FIGURE 1. SUB1, SUB2, SUB3 in root




```
      .ROOT  ROOT-*(A-XX,B-XX,C-XX)
XX:   .FCTR  *(SUB1,SUB2,SUB3)
      .END
```

FIGURE 2. Overlaying SUB1, SUB2, SUB3




```
      .ROOT  ROOT-*(A,B,C),USEROV
@USRLIB.ODL
      .END

USRLIB.ODL:

      .NAME  NULL
USEROV: .FCTR  NULL-*(SUB1,SUB2,SUB3)
USERAL: .FCTR  SUB1-SUB2-SUB3
```

FIGURE 3. Using a cotree

Software Product Description

Product Name: LOCK-11 Version 2.3

Description:

Lock-11 is a security superstructure built upon the standard RSTS password structure that provides the following extensions:

- Full V7.1 support
- USERID implementation for Dial-in, Packet and Concentrators.
- Macro DPY — a rewrite of VT5DPY for VT100's, interfaced with security file and featuring a repertoire of 'UT' type commands for system managers.
- First release of SAFE-11, a 'menu' environment (KBM/RTS) that keeps non-privileged users where they belong.
- Absolute control of system access by keyboard. Manager may limit any keyboard to certain accounts or groups of accounts and control time as well as day of week access.
- Password knowledge is no longer carte blanche system access. System detects unauthorized use of passwords. Privileged passwords don't work on non-privileged keyboards. Non-privileged passwords work only on specified keyboards.
- Real time system surveillance. Manager specifies a list of alarm keyboards which log all infractions and probes as they happen. Opser is not required.
- Auto-login (with or without password) and chain with specified core common contents by KB.
- Manager may establish special priority/burst settings by KB. Manager may establish default output protection code, @ assignment and up to three specific user logicals for each KB. Default RTS is also selectable. All assignments are made at log-in.
- Manager specifies a list of console keyboards from which security file editor may operate.
- Manager may define a KB-specific access-denied message.
- Manager may specify number of retries before access-denied and number of access-denied messages before line disable. Hangup on access denied is optional. All above may be specified on a per-kb basis.
- A macro DYNPRI program is included which performs the following functions:
  - Users may be dispatched into ten separate priority queues, separately tunable on-line. Each queue has ten levels. Queues are selectable by KB.
  - DYNPRI detects hibernating jobs and announces the fact on ALARM keyboards. Privileged jobs hibernating cause extra loud and long alarms.
  - DYNPRI produces almost no load in operation and runs in 5K words.
  - DYNPRI will hold up to fourteen files open for performance purposes.
  - DYNPRI will kill hibernating jobs in up to ten [p,*] or [p,pn]'s

Minimum Hardware/Software Required:

Any valid RSTS/E system running Version 7.0 or later. Any version of RSX emulation is needed.

Support: See License Agreement

Installation: User Installed

Ordering Information:

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

First license	0% discount
Second thru Fourth license	40% discount
Fifth thru Twentieth license	70% discount

Licensed users desiring source code for internal use only must execute a separate Program Sources License Agreement. Sources are available at ten times the initial license fee.

License Fee:

Single CPU license: $1250.00. Annual maintenance at 12% of current list price.

Contact:
Dave Mallery
Nationwide Data Dialog
215—364-2800

CIRCLE 12 ON READER CARD

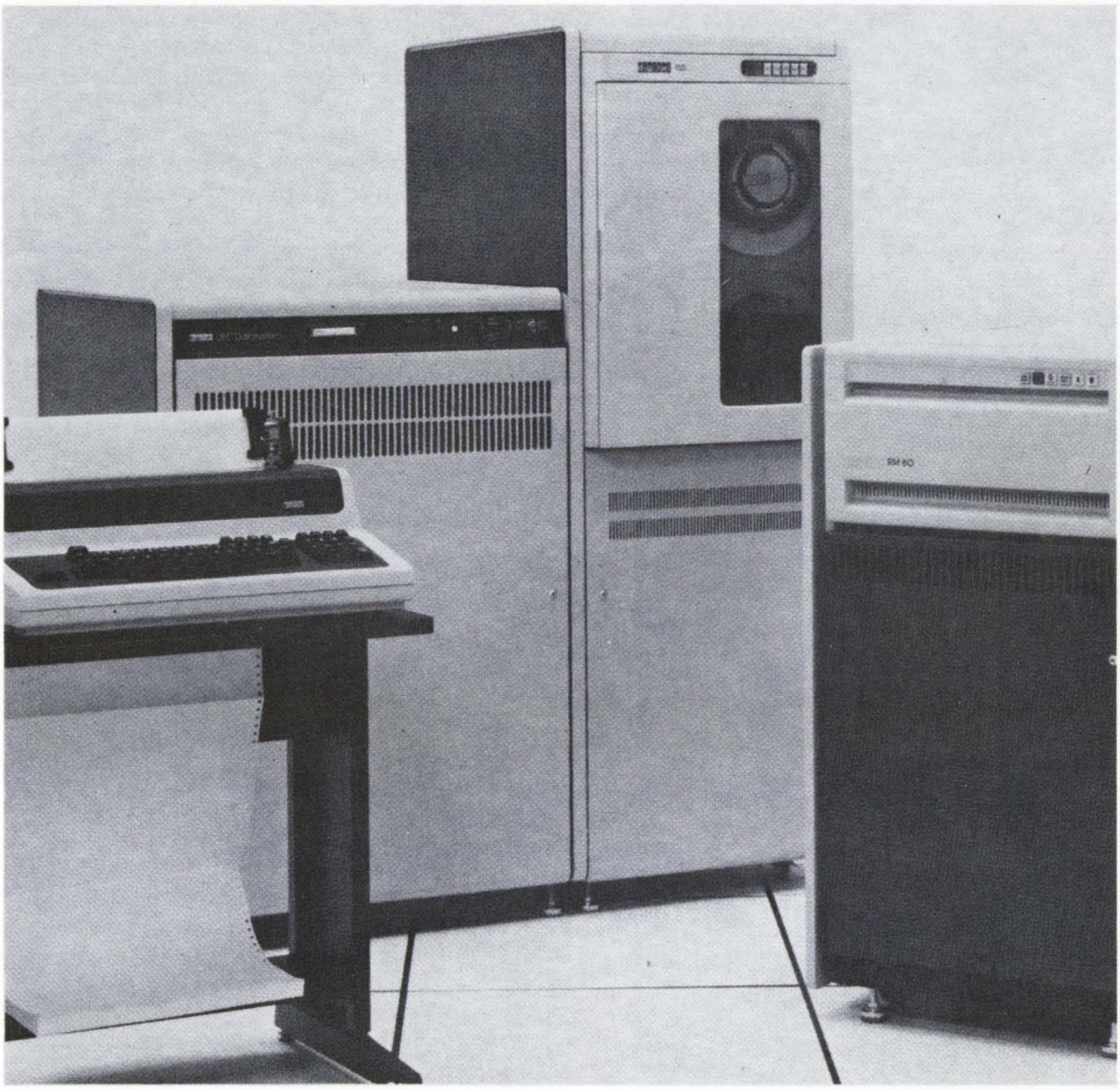
The RSTS CRYSTAL BALL - Part 1

. . . continued from page 33

```
.ENABL  LC
TITLE   XPEEK,<Disassembled .XPEEK code>,11,26-Apr-82,MCG

;
; The following is hand-disassembled code from RSTS V7.1-11
; Sorry, I don't have REAL sources ...
;

;+
; .XPEEK - Extended block mode PEEK
;
;       R1 -> JDB @ JDFLG
;       R3 -> (Mapped) User's XRB
;
;               + XRLEN         = Count of bytes to move
;               + XRBC          = Must be zero
;               + XRLOC         = Buffer location
;               + XRBLKM        = MSB of physical address
;               + XRBLK         = LSB of physical address
;
;       On return
;
;               Block of memory moved to user's buffer
;
;               -or-
;
;               ERROR   PRVIOL
;               ERROR   BADCNT
;
;-

        EMTENT  .XPEEK

.ENABL  LSB

        BIT     #JFPRIV!JFSYS!JFNOPR,(R1)  ;Is abuser priviliged?
        BEQ     50$             ; nope, that's most certainly an error
        MOV     (R3),R0         ;Pick up byte count to move
.ASSUME XRLEN   EQ      0
        CMP     R0,#20000-200+1 ;Can we map the whole buffer in PAR6?
        BHIS    60$             ; nope, error
        CALLX   CHKXRB          ;Verify XRB ok for read (C=1, BHIS=BCC)
        MOV     R0,XRBC(R3)     ;Duplicate the count now
        MOV     XRBLK(R3),R5    ;Pick up least significant word of address
        MOVB    XRBLKM(R3),R4   ;Now get the MSB of address (6 bits, anyway)
        BMI     50$             ;PDP-11's don't have 24 bit addressing!
        MOV     R4,R2           ;Copy the
        MOV     R5,R3           ; physical address
        MOV     R5,R1           ;Copy the LSW
        BIC     #77,R1          ;Round down to MMU boundary
        ADD     #120000,R1      ;Offset into DPAR5 for posterity
        ASHC    #-6,R4          ;Convert physical address to MMU units (1 word)
        DEC     R0              ;An appropriately named instruction ...
        ADD     R0,R3           ;Offset the physical address
        ADC     R2              ; to the end of the requested segment
        INC     R0              ;Un-fudge byte count now
        ASHC    #-6,R2          ;Convert ending address to MMU units (1 word)
        MOV     #MEMLST,R4      ;Pick up root of MEMLST

10$:    MOV     M.PHYA(R4),R2   ;Get physical address of this entry
        CMP     R2,R3           ;Past end of requested block?
        BHI     40$             ; yep, ok to move it then
        TST     M.PNXT(R4)      ;Are we at the end of MEMLST?
        BEQ     50$             ; yep, that's an error
        SUB     R5,R2           ;In specified area of memory?
        BCC     20$             ; completely ...
        NEG     R2              ;Form amount over specified start
        ASH     #-5,R2          ;Convert for compare
        CMP     R2,M.TSIZ(R4)   ;Fit in this block?
        BHIS    30$             ; nope, don't check it for validity
        CMPB    R2,M.SIZE(R4)   ;Really-really fit in this block?
        BHIS    30$             ; naw, not really-really

20$:    MOV     R4,R2           ;Copy address of memory sub-block
        BIC     #^C<37>,R2      ;Clear all but location within a control block
        BIT     BITWRD(R2),#14  ;Is this memory part of a RTS or locked out?
        BNE     50$             ; yep, sorry jack (Steve?) ...

30$:    MOV     M.PNXT(R4),R4   ;Get the next pointer in the list
        BR      10$             ; and loop

40$:    MOV     R0,R2           ;Copy count of bytes to move
        MOV     R5,R0           ;Save physical source address
        MOV     @#140000+XRB+XRLOC,R5  ;Get abuser's buffer location again
        CALLX   SCRUMP          ; and map it in PAR6
        MOV     R5,R3           ;Copy (biased, virtual) buffer pointer
        CALLRX  MOVMEM          ;Now off to move that memory!

50$:    ERROR   PRVIOL          ;Can't XPEEK locked out memory,
                                ;run-time systems, resident libraries,
                                ;non-existant memory, dead wombats,
                                ;black holes, etc ...

60$:    ERROR   BADCNT          ;Illegal XPEEKing parameters ...

.DSABL  LSB

GLOBAL  <MEMLST,BITWRD>
.END
```

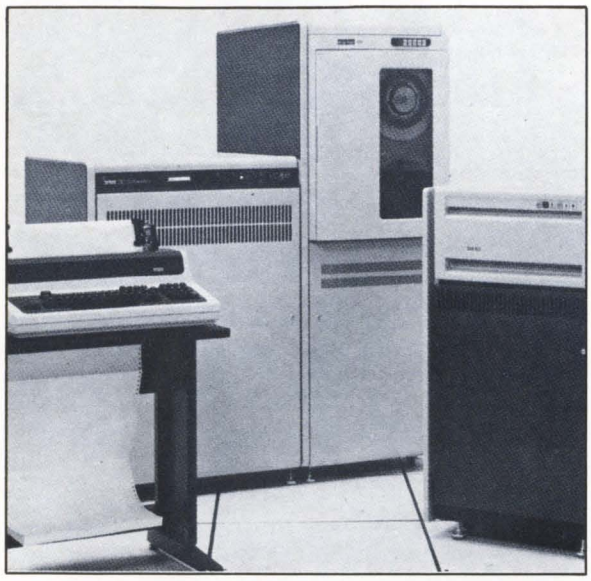
The VAX-SCENE

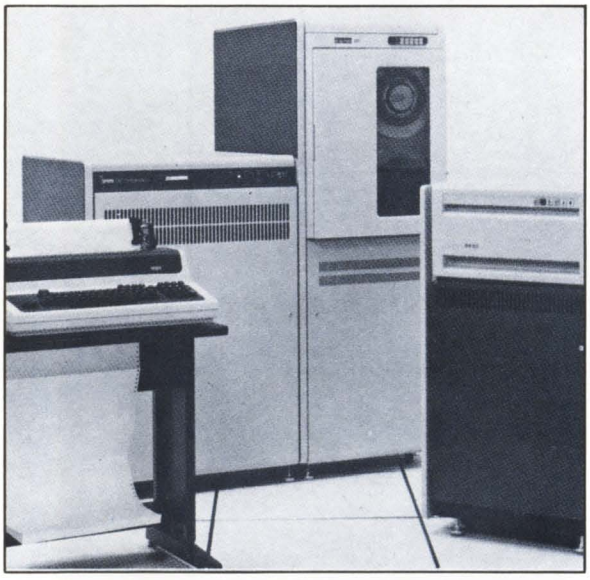
Number 8 (RSTS PROFESSIONAL, Vol. 4, No. 3) June 1982

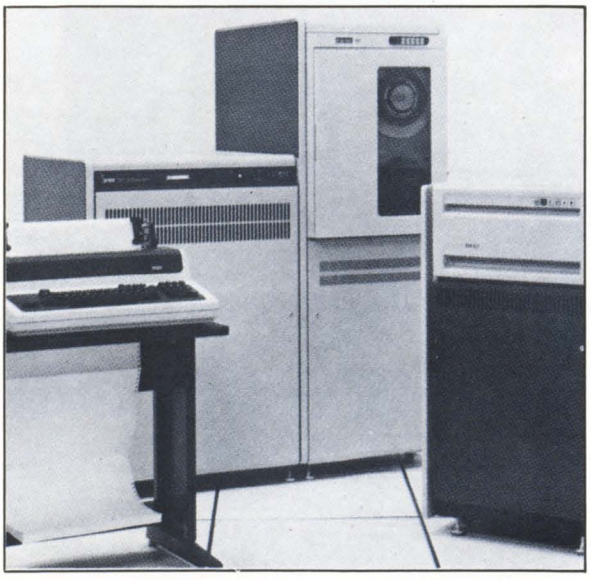


INSIDE:

- ☐ **Word Processing for the VAX**
- ☐ **A File Compress Utility for VAX/VMS Systems**







WORDPROCESSING FOR THE VAX

By Eric S. Dickman, President, EEC Systems

VAX-11, A TECHNOLOGICAL BREAKTHROUGH

When DEC announced the VAX-11 computer, its 32 bit addressing capabilities were hailed as a technological breakthrough. Moreover, DEC themselves have made it clear that although the 16 bit PDP-11 range of computers will receive a good portion of their R&D dollars in the 1980's, it is the VAX family of computers to which DEC intends to direct its best developmental efforts. New members of the VAX family are currently being designed at DEC, so that by the 1990's, reliable sources inside DEC predict that the VAX will represent the major thrust of DEC's product line.

VAX-11 THE IDEAL CHOICE FOR W-P

When considering the new purchase of a computer which would be suitable for O/A and W-P, the VAX computer presents an obvious choice, but only if the initial large capital outlay does not present a serious drawback. The reason why VMS is such an excellent choice for W-P, is that it is a user-friendly operating system which has been designed in conjunction with the VAX-11 hardware for an interactive environment. It also has some useful utilities such as 'talk' and 'mail' allowing for interbuilding communications or support questions to be directed quickly to the system manager. However, the initial outlay for the hardware and operating system only makes the choice of the VAX viable for installations of between 32 and 64 users, at which point the cost per work station becomes more cost effective than an upper end PDP-11. Also consider at the low end, DECMATE II to be released in Summer 1982 at the rumored price of $5,000 which includes LQP, a VT100, Floppy Disk, Processor and choice of W-P software. At these prices W-P on the VAX is only financially competitive for a large number of users with needs for large storage. It should also be mentioned that at the current time there is no word-processing software which runs effectively with 64 concurrent W-P users on a VAX. This is because W-P is particularly demanding on a system's resources, although there is a great deal of variability in the speed and efficiency of W-P software currently available. However, since most installations are not dedicated to W-P and are used for a variety of other different kinds of jobs, the VAX computer may still represent an excellent choice; especially since DEC have recently announced some high quality O/A software for the VAX, such as DEC set and DEC mail, which together with some new office management software due to be released in Summer 1982 represent a near comprehensive integrated O/A system that only lacks for W-P. For a schematic diagram of wordprocessing, see Figure 1.

COMPATIBILITY VERSUS NATIVE MODE

Most of the wordprocessing software packages for the VAX are currently available in what is known as 'compatibility' mode rather than 'native' mode. W-P software running in native mode is substantially faster and the reasons for this need some explanation. When the VAX-11 series was designed, it was obvious from the marketing point of view that these new processors had to have a well-defined growth path from the older PDP-11 processor series. This was facilitated by the fact that the principal designer/implementer of the VAX series micro code and VMS operating system was in fact the originator of the RSX operating system family. Therefore, VMS was given an Applications Migration Environment Monitor which originally ran under an RSX executive to be executed in instruction compatibility mode under VMS. Obviously if the VAX processor has to emulate a foreign instruction set, it will use a proportion of its power to perform the emulation during code execution. This means that code executing in compatibility mode is not as efficient as code executing in native mode on the VAX. For comparison purposes it is said that the processor power of a VAX running in compatibility mode is slightly more powerful than a PDP-11/70. Of course the VAX has a much higher band-width I/O bus and so will seem to be faster than an 11/70 when used in compatibility mode. We ourselves at EEC Systems found that our LEX-11 W-P software ran concurrent 20-25 users comfortably without significant degradation on a VAX in compatibility mode. The new native mode version of LEX-11 being released in late Summer 1982 is predicted to run between 5-7 times faster than the compatibility version. While this conversion was not a trivial task it was undoubtly easier than for some W-P software packages. This is because the operating system dependent parts of LEX-11 are located in one module with a common interface to the other modules of LEX-11. The user interface is defined by the operating system independent modules.

POINTERS WHEN PURCHASING W-P

Beyond these considerations mentioned above there are other more general points about W-P software, that the purchaser of W-P for a VAX computer should bear in mind. Since many software packages appear on the surface to provide the user with the same functionality, it is only by taking a closer look that some of the finer but important differences become apparent.

USER-FRIENDLY FEATURES

Systems are often dubbed as 'user-friendly', but what does that term mean when you get down to brass tacks? There are probably two crucial features from the users point of view. One involves being able to accomplish most of the common wordprocessing functions with a single keystroke; the other is the ability to use English language type of commands and not have any hidden embedded control characters or visible W-P commands in the text. Or to put it another way, 'What you see is what you get'. With some W-P systems it is impossible to see the finished form of a document until it has been through a 'RUNNOFF TYPE' post processing operation. In practical terms this means that corrections, be it editing, margins or pagination can only be corrected and executed in an additional operation which adds a considerable amount of time to the W-P process.

FLEXIBILITY

'Flexibility', another catch-phrase in wordprocessing covers a number of things. Only some systems have a true full screen editor allowing for easy cursor movement and editing any place on the screen of text. This is not only kinder on the eyes of the operator, but speeds up the editing process. Most systems cannot move text or columns horizontally as well as vertically, changing the order for instance of columns, used with financial applications. Text entry methods should have the capability for true multiple column text entry. These columns can be displayed side by side on the screen using rulers, and right and left margin markers to delineate column parameters. Column entry is especially useful when using a calculator feature, and if the calculator is an integral part of the W-P program, text and calculations may be entered without changing the mode. If a W-P package has keystores these can hold the equivalents of all functions executed so that any application can be automatically executed. This can be useful when used in conjunction with a calculator for automatic invoice production.

If the W-P package allows for storing of sequences of keystores that might be needed for later recall and if that ability is coupled with some conditional abilities as to either the existence of documents or the existence of strings within documents, then the package virtually contains a programming language. Programmers can also edit or create their programs using the W-P software, which allows them to use the W-P editing and recall features. Compilation errors would not be a source of users concern if the software did not use hidden embedded characters. VAX users typically use their machines for both data processing and W-P functions, so that the better W-P software is capable of being used by secretaries and programmers alike.

Flexibility can also apply to whether the user can customize the software for his own application. So-called 'soft-coded' W-P packages are to be preferred and are defined as ones where input and output interfaces may be defined or modified by the end user. This means that the user only has to specify the control sequences required in order to use any of a wide variety of terminals and their special facilities, such as function keys. Users can thus tailor the W-P package to their own application requirements and change the functions of the keys. User customization can also be accomplished if the menus can be changed or added, particularly useful when modifying a package to suit a particular business environment. If W-P software produces standard ASII text files without any hidden control characters coupled with a return facility, then external programs can be hooked into the W-P software and appear to the user as if they are part of the W-P software itself, which is accomplished just by adding another menu option. For the VAX user planning a complete O/A system, or using existing large database it would be thus important to ascertain whether the W-P software chosen can be integrated with other external software. We have found that a feature of W-P systems that is appreciated in particular by users is the availability of a forms sys-

tem. In these systems, users can design the layout and generate their own forms, storing the blank forms away for later recall. Since VAX users in particular seem to need to use W-P .P software for a variety of different purposes, the ability of the software to be adapted to respond to the needs of the various users should not be underrated. When new users of W-P have passed the hurdles of initial training, they begin to appreciate the potential uses of W-P software, providing of course that the software can be easily modified to fit their application.

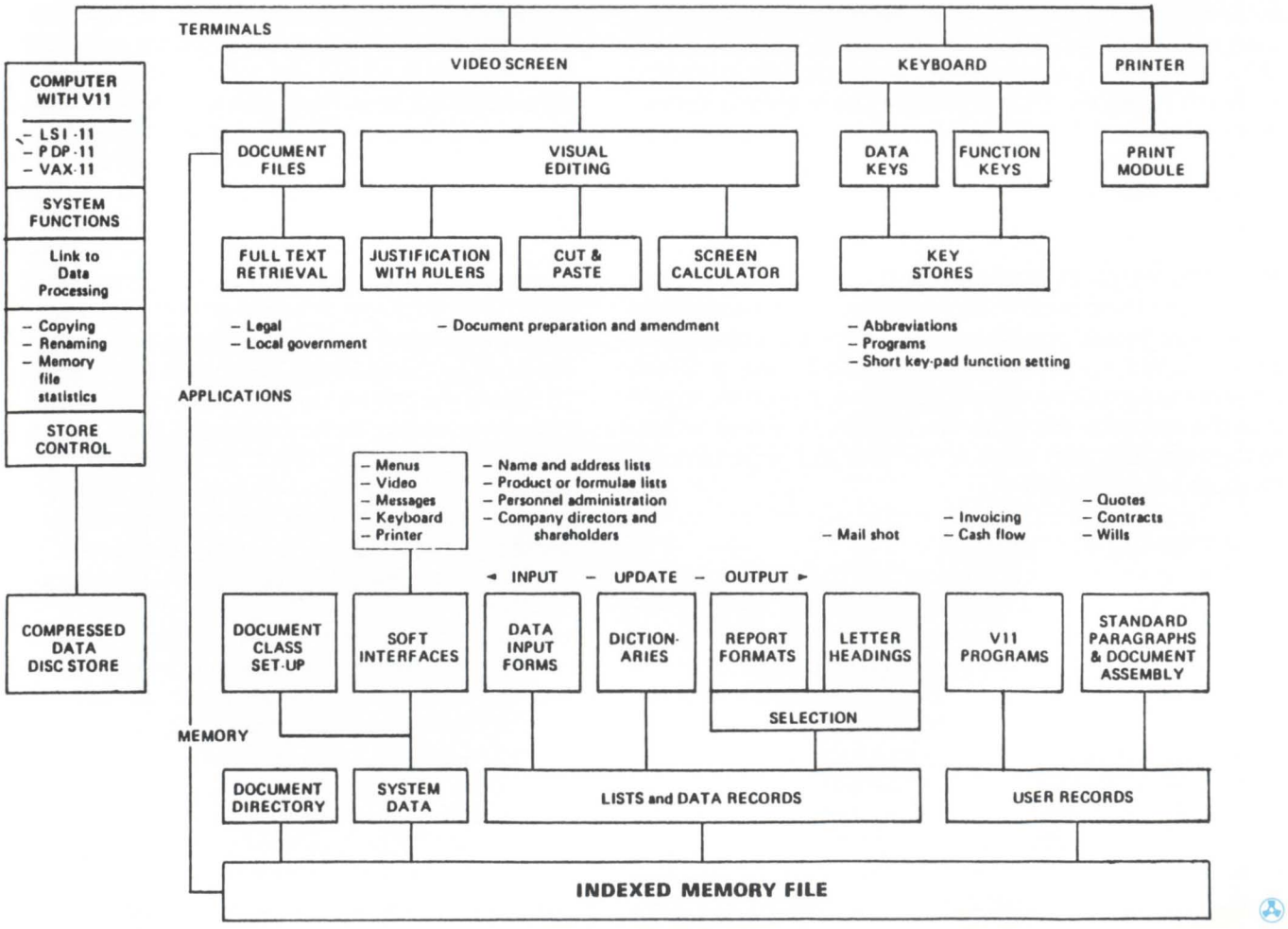
We have found that VAX users want more than plain vanilla W-P software. The better packages on the market thus provide some integration of W-P with database management and list processing. If system records are stored index sequentially, then they can be retrieved much faster than systems which search through the entire database in order to access a particular record. If a system allows for the efficient storage of a variety of additional information besides name and address in mailing list applications, then a sort and select feature which incorporates Boolean operators can select out names from a master file on any criteria. For instance, when keeping a file of customers, reports should be able to be easily and quickly generated by type of product, the month in which sales were made, a particular geographical area, etc. Another good auxiliary program creates and stores standard paragraphs; the memory file can be directly accessed and hold data records of varying lengths. In sum, wordprocessing for VMS is a lot more sophisticated than simple text editing.

FUTURE TRENDS

We now have a situation in which wordprocessing software under VMS is available and already there is software which is comparable and even superior in some cases to that found on the better stand-alone products. In the future, it is likely that with the predicted 'UNIX Revolution' that VAX users may have an increased need for W-P running under the UNIX operating system. While UNIX on the VAX seems at the moment to be largely confined to universities, the recent significant decrease in the price of the UNIX operating system has opened up new markets for UNIX in the business world. Many industry observers predict that UNIX may soon become the industry standard, overtaking CP/M. Although DEC have refused to support or market UNIX at the moment, that situation may change as the UNIX movement gathers momentum.

Eric Dickman is the President of EEC Systems, Wayland, Masachusetts. He holds a masters degree in Engineering and an MBA from Cornell. He was employed in Strategic Planning in Digital Equipment Corporation before leaving in 1980 to form his own company. EEC Systems markets LEX-11 wordprocessing software.

WORD PROCESSING SCHEMATIC DIAGRAM



DEC USERS

Test your word processing I.Q.

And discover how the advanced features of LEX-11 can increase your w-p capabilities.

THE CHALLENGE:

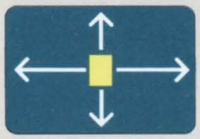
Almost any word processing software package will have such features as menus, editing, spelling error detection, list processing, cut and paste, automatic word wrap and automatic pagination. . .

. . .**But** can you tell the difference between run-of-the-mill software and a system that is really special?

EEC SYSTEMS offers you this challenge! Test your word processing I.Q.

THE QUESTIONS:

1 EDITING



Which kind of editing operation is quickest to execute and easiest on the eyes of the word processing user?

a) Full screen editing allowing for easy cursor movement around the screen?

b) Moving the cursor around by doing a line count?

c) Editing on the bottom line of text only?

2 DOCUMENT LAYOUT

Whatever document format you choose. . .you want to see what the finished article will look like. Should you. . .

a) View it on the screen as it would come out of the printer?

b) Run it through a pre-processor to see what it looks like and then if you like it, print it?

3 KEYSTROKES

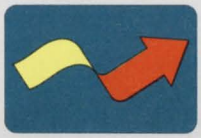
Using a well designed w-p system, how many keystrokes should it take to execute the most often used w-p functions?

a) One easy stroke with no codes?

b) Two or more with complex w-p codes?

c) Three or more?

4 FLEXIBILITY

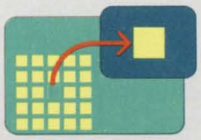


As the business manager of your company, you would like to find w-p software that you can tailor to your company's specific needs. Should you. . .

a) Look for w-p software that allows you to change and add menus, and change function keys?

b) Write your own custom software?

5 RETRIEVAL



If you want to retrieve information quickly from a large database, which w-p software should you choose?

a) One that can access a particular record by going to it directly?

b) One that searches through all the records on the database sequentially until it finds the right one?

6 COMPATABILITY

As a manager of MIS, you want a w-p system that can be integrated with any other DEC compatable application software. Should you choose w-p software with. . .

a) ASCII formated files?

b) Software which requires non-printing characters in it's file system?

7 MATH

Your company has a number of financial applications and is looking for a w-p package with math capabilities. Should you choose. . .

a) On screen calculating allowing for editing, storing and recall of equations, calculations integrated with your word processing applications?

b) Software where the math capabilities are tied to the list processing module?

c) A separate math package?

THE ANSWERS:

If you answered "a" to all of the questions above, go to the top of the class. Chances are that you already use **LEX-11** or are about to buy it. You know that **LEX-11** is the **ONLY** word processing software that combines these advanced features. . .and more. . . in one software package.

PUT OUR SOFTWARE TO THE TEST

Call for a demonstration by modem in your office today.

(617) 358-7782 **(617) 443-6376**

EEC SYSTEMS INC.

Dept. TDP , 286 Boston Post Road, Wayland, MA 01778

LEX-11 is available under the following operating systems: RT-11, TSX-Plus, RSX-11M, RSTS/E, IAS, UNIX (both PDP-11 and VAX), IDRIS, VMS. . . **and** it works in exactly the same way, with the same features for all operating systems, with your existing peripheral equipment.

CIRCLE 43 ON READER CARD

A FILE COMPRESS UTILITY FOR VAX/VMS SYSTEMS

By Andrew G. Gault, Transcomm Data Systems Incorporated

Transcomm Data Systems Incorporated is an Authorized DIGITAL Computer Distributor specializing in business software packages. Since 1972 Transcomm has worked exclusively with DEC equipment specializing initially in PDP-11 RSTS/E systems. In 1980 Transcomm began to address the special needs of the new 32 bit VAX machines.

One area of concern on VAX/VMS systems was a severe shortage of disk space. This problem became critical very quickly on the disks used for product development and maintenance. Disk compresses (DSC-2) were done on a regular basis but with no improvement.

The critical factor was that some DEC disk utilities use the allocated disk space and not the actual space. After many years of working with RSTS/E systems it was assumed that a disk compress would, in addition to making the files contiguous, reallocate disk space in the most efficient manner. This was a false assumption. An improperly allocated file will continue to waste disk space.

Since all VAX/VMS files are RMS files, this procedure of using the allocated disk space as the minimum required disk space makes sense. But it is of little consolation to the user who is continually running out of space.

In a normal production environment this problem would not occur very frequently. The files are relatively stable and any highly active files are usually deleted shortly after they are created. It is in the program development and maintenance areas that the problem becomes critical.

For example, when files with program updates are created (i.e., patch files) the source program is used as the base code. The source code is updated and placed into the distribution account. All code but the patch is deleted and then stored in a patch file for distribution to installations with custom modifications. Because this patch file was created from the source code it is allocated the same amount of space as the source code. The patch files are usually smaller than the source code so that much of the allocated space is wasted. Therefore, if a source program of 130 blocks is patched and the patch file has a size of 10 blocks it will be allocated 132 blocks instead of 12 blocks (cluster size = 4).

One solution is to copy the file into a temporary file, delete the original, then rename the temporary file as the original thus preserving the version number. This method is fine for a small number of files on a particular account but is tedious when a full disk is involved. This presents several obstacles. Previous generations for a file should be purged, and directory (*.DIR) files which can not be copied without losing file pointers.

To simplfy this procedure Transcomm developed a DCL utility to compress files by a specified disk, account, and file name combination. The utility proceeds:

ENTER DISK NAME (I.E. DRA1:) >

Enter a logical disk name.

Depress the RETURN key to use the current device as the default.

Enter /E to exit the utility. The use of /E is a Transcomm convention which allows the user to branch back one prompt level.

ENTER ACCOUNT IN THE FORM [X.Y.Z] >

Enter an account name, wild cards are allowed.

Depress the RETURN key to use the user's default account.

Enter /E to return to the ENTER DISK NAME prompt.

ENTER FILE NAME(S) (CR = *.*) >

Enter a file name or string of file names, wild cards are allowed.

Depress the RETURN key to use the wildcard specification *.* which will find all files on the given account(s).

Enter /E to return to the ENTER ACCOUNT prompt.

USE /LOG ON ALL COMMANDS (Y/N = CR) ?

Enter Y to append the /LOG option to all copy and purge commands used in the file compress.

Enter N or depress the RETURN key so no system log messages are displayed during the file compress.

Enter /E to return to the previous prompt.

At this point, the utility has all the information required to search the disk and produce a directory of all files which meet the specifications. However, as mentioned previously, multiple generations of a file may be undesirable and, therefore, the disk should be purged before the directory file is built. So the utility prompts:

START PURGE (Y/N = CR) ?

Enter Y to purge the disk using the file paramters entered in response to the first set of prompts.

Enter N or depress the RETURN key to cancel this procedure and retain all versions.

Enter /E to return to the previous prompt.

Using the file name specifications entered by the user, the utility builds a directory file with all the requested file names excluding *.DIR files. Each record in this directory file is read. From the record the logical device, account and file name are removed. This information is then written into a command file which will do the actual copy/purge sequence. When all file names have been read the directory file is deleted and a message is displayed.

The utility prompts the user:

COMMAND FILE READY, ORGANIZE (Y/N = CR) ?

Enter Y to execute the command file as an indirect command file. When the compress is finished the command file is deleted and the user is returned to the ENTER DISK NAME prompt.

Enter N or depress the RETURN key to save but not execute the command file. This allows the user to check the command file and then execute it as a separate process. The user is returned to the ENTER DISK NAME prompt.

Enter /E to abort the process and return to the ENTER DISK NAME prompt.

It is highly recommended that a disk compress (DSC-2) be done after this file compress utility is finished, since the file compress will leave the disk space fragmented and using the disk will slow the system down. A disk compress (DSC-2) will correct this situation and organize the disk files in a contiguous form.

Proper use of this utility has solved many space problems at Transcomm. Constructive criticisms and suggestions are welcomed by the author at (412) 963-6770.

```
$       !*****************************************************************
$       !                                                               &
$       !       SYSTEM:         VAX OPERATIONS                          &
$       !       MODULE:         DISK SUPPORT                            &
$       !       PROGRAM:        DSKCMP.COM                              &
$       !       VERSION:        V4.00                                   &
$       !       DATE:           82/04/15                                &
$       !       AUTHORS:        ANDREW G. GAULT                         &
$       !                       VINCE SPADARO                           &
$       !                       DAVE FROBLE                             &
$       !                                                               &
$       !       PURPOSE/DESCRIPTION: DISK FILE COMPRESS COMMAND UTILITY  &
$       !                                                               &
$       !       SOME OF THE VAX/VMS UTILITIES AND COMPILERS ALLOCATE MORE &
$       !       STORAGE TO A DISK FILE THAN IS REQUIRED.  AS A RESULT, A  &
$       !       PORTION OF THE DISK SPACE IS WASTED WITHOUT THE USER BEING &
$       !       AWARE OF THE LOSS.  THEREFORE, ON A HIGHLY USED DEVICE SUCH &
$       !       AS DEVELOPMENT AND PROGRAM MAINTENANCE A GREAT DEAL OF THE &
$       !       ALLOCATED BLOCKS ARE WASTED.                            &
$       !                                                               &
$       !       THIS PROBLEM CAN BE DETECTED BY DOING A DISK DIRECTORY USING &
$       !       THE /SIZE=ALL OPTION.  IF A VARIANCE OF MORE THAN THE DISK &
$       !       CLUSTER SIZE LESS ONE EXISTS BETWEEN THE ACTUAL AND ALLOCATED &
$       !       BLOCKS THE FILE HAS BEEN ALLOCATED AN EXCESS AMOUNT OF  &
$       !       STORAGE.                                                &
$       !                                                               &
$       !       ONE SOLUTION IS TO COPY THE FILE INTO A TEMPORARY FILE, DELETE &
$       !       THE ORIGINAL VERSION, THEN RENAME THE TEMPORARY FILE AS THE &
$       !       ORIGINAL THUS PRSERVING THE VERSION NUMBER.  DSKCMP.COM IS A &
```

```
$       !       'DCL' UTILITY PROGRAM DESIGNED AT TRANSCOMM DATA SYSTEMS TO    &
$       !       PERFORM THIS COPY/DELETE/RENAME SEQUENCE FOR A SPECIFIED DISK, &
$       !       ACCOUNT, AND FILE COMBINATION.                                 &
$       !                                                                      &
$       !       IMPORTANT NOTES :                                              &
$       !                                                                      &
$       !       *.DIR FILES ARE NOT PROCESSED.                                 &
$       !                                                                      &
$       !       NO USERS SHOULD BE LOGGED INTO THE ACCOUNT(S) BEING            &
$       !       PROCESSED.                                                     &
$       !                                                                      &
$       !       A DISK COMPRESS (DSC-2) SHOULD BE RUN AFTER EXTENSIVE USE OF   &
$       !       THIS UTILITY IN ORDER TO PREVENT FRAGMENTATION OF THE DISK.    &
$       !                                                                      &
$       !       THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT  &
$       !       NOTICE AND SHOULD NOT BE CONSTRUCTED AS A COMMITMENT BY        &
$       !       TRANSCOMM DATA SYSTEMS INC.                                    &
$       !                                                                      &
$       !       THIS SOFTWARE IS UN-RELEASED AND TRANSCOMM DATA SYSTEMS INC.   &
$       !       HAS NO COMMITMENT TO SUPPORT IT.                               &
$       !                                                                      &
$       !**********************************************************************&
$       !                                                                      &
$       !       I/O CHANNEL SPECIFICATIONS:                                    &
$       !                                                                      &
$       !       -CH VAR--------FILE VAR------DESCRIPTION------                 &
$       !        DIRFIL        'DISK'.DIR    DIRECTORY FILE                   &
$       !        CMDFIL        'DISK'.COM    COPY/PURGE COMMANDS              &
$       !        SYS$OUTPUT    SYS$OUTPUT    KB:                              &
$       !                                                                      &
$       !**********************************************************************&
$       !                                                                      &
$       !**********************************************************************&
$       !                                                                      &
$       !         COPYRIGHT 1982 BY TRANSCOMM DATA SYSTEMS INCORPORATED        &
$       !              1380 OLD FREEPORT RD., PITTSBURGH, PA  15230            &
$       !                                                                      &
$       !       THIS SOFTWARE IS FURNISHED FREE OF CHARGE TO MEMBERS OF THE    &
$       !       NORTH AMERICAN DECUS ORGANIZATION AND MAY BE COPIED ONLY WITH  &
$       !       THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR &
$       !       ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE &
$       !       AVAILABLE TO ANY OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF   &
$       !       THE SOFTWARE IS HEREBY TRANSFERRED.                            &
$       !                                                                      &
$       !**********************************************************************&
$       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< INITIALIZATION >>>>>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ BANNER:                                               !  PRINT THE BANNER:
$       WRITE SYS$OUTPUT ""
$       WRITE SYS$OUTPUT "Transcomm Disk File Compress Command Utility"
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< PROMPT FOR FILE SPECS >>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ DISK_IN:                                              !  GET THE DISK'S
$                                                       !  LOGICAL NAME:
$       WRITE SYS$OUTPUT ""
$       INQUIRE DISKMN "Enter Disk Name (i.e. DRA1:) "  !  PROMPT THE OPERATOR
$               IF DISKMN .EQS. "/E" THEN GOTO CMP_END  !  END THE PROCESS IF
$                                                       !  /E WAS ENTERED
$       DISKMN_LEN = 'F$LENGTH( DISKMN )'               !  GET THE LENGTH OF
$                                                       !  THE ENTRY
$       DISKDIR := 'DISKMN'                             !  STORE THE ENTRY
$                                                       !  IN ANOTHER VARAIABLE
$                                                       !  WHICH IS USED IN THE
$                                                       !  DIRECTORY COMMAND
$                                                       !  THIS IS NEEDED IN
$                                                       !  CASE OF A NULL
$                                                       !  ENTRY
$       IF DISKMN_LEN .EQ. 0 THEN DISKMN := "DMAGN:"    !  IF NO ENTRY THEN SET
$                                                       !  THE DEVICE NAME TO A
$                                                       !  DUMMY FOR THE FILE
$                                                       !  NAMES
$       DISKMN_COL = 'F$LOCATE( ":" , DISKMN )'         !  SCAN FOR THE COLON
$       IF DISKMN_COL .EQ. DISKMN_LEN THEN DISKMN := 'DISKMN':
$                                                       !  ADD THE COLON IF IT
$                                                       !  WAS NOT FOUND
$       DISKMN_COL = 'F$LOCATE( ":" , DISKMN )'         !  FIND THE COLON
$                                                       !  AGAIN SINCE IF IT
$                                                       !  WAS ADDED THE OLD
$                                                       !  OFFSET IS WRONG
$       DISK := 'F$EXTRACT( 0 , DISKMN_COL , DISKMN )   !  PULL OUT THE DISK
$                                                       !  NAME WITHOUT THE
$                                                       !  COLON SO THIS CAN
$                                                       !  BE USED IN FILE
$                                                       !  NAMES
$                                                       !
$ ACCT_IN:                                              !  PROMPT FOR A
$                                                       !  SPECIFIC ACCOUNT:
$       INQUIRE ACCTMN "Enter Account In The Form [X.Y.Z] "
$                                                       !  PROMPT THE USER FOR
$                                                       !  AN ACCOUNT, DEFAULT
$                                                       !  IS ALL ACCOUNTS
$               IF ACCTMN .EQS. "/E" THEN GOTO DISK_IN  !  BRANCH TO THE DISK
$                                                       !  PROMPT IF A /E WAS
$                                                       !  ENTERED
$                                                       !
$ FILE_IN:                                              !  PROMPT FOR A STRING
$                                                       !  OF FILE NAMES:
$       INQUIRE FILEMN "Enter File Name(s) ( CR=*.* ) " !  PROMPT THE USER FOR
$                                                       !  THE FILE NAMES
$               IF FILEMN .EQS. "/E" THEN GOTO ACCT_IN  !  ON /E BRANCH TO THE
$                                                       !  PREVIOUS PROMPT
$       IF FILEMN .EQS. "" THEN FILEMN := "*.*          !  IF A NULL STRING
$                                                       !  SET THE FILE NAME
$                                                       !  VARIABLE FOR ALL
$                                                       !  FILES
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< LOG OPTION >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ OPTION_IN:                                            !  SHOULD /LOG BE USED
$                                                       !  AS AUDIT TRAIL:
$       LOG := ""                                       !  SET VARIABLE TO NULL
$                                                       !  SINCE DEFAULT IS NO
$       INQUIRE LOGMN "Use /LOG On All Commands ( Y/N=CR ) "
$                                                       !  PROMPT THE USER FOR
$                                                       !  A Y/N RESPONSE
$               IF LOGMN .EQS. "/E" THEN GOTO FILE_IN   !  ON A /E BRANCH TO
$                                                       !  THE PREVIOUS PROMPT
$               IF LOGMN THEN LOG := "/LOG"             !  IF YES THEN RESET
$                                                       !  VARIABLE FOR /LOG
$                                                       !  AUDIT
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< PREPARE THE DISK >>>>>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
```

```
$ PRE_CLEAN:                                            !  MAKE SURE THE DISK
$                                                       !  IS CLEAN:
$       WRITE SYS$OUTPUT ""                             !  DISPLAY WARNING
$       WRITE SYS$OUTPUT "A Purge Of Old Versions Is Recommended Before"
$       WRITE SYS$OUTPUT "The Directory File Is Built"
$       WRITE SYS$OUTPUT ""
$       INQUIRE VERIF "Start Purge ( Y/N=CR ) "         !  PROMPT THE USER
$                                                       !  TO VERIFY THE
$                                                       !  START
$               IF VERIF .EQS. "/E" THEN GOTO OPTION_IN !  ON /E BRANCH TO
$                                                       !  THE PREVIOUS
$                                                       !  PROMPT
$               IF VERIF THEN GOTO PRE_PURGE            !  ON Y THEN BRANCH
$                                                       !  TO DO PURGE
$       WRITE SYS$OUTPUT ""                             !  OTHERWISE TELL THE
$       WRITE SYS$OUTPUT "Purge Not Performed"          !  USER THAT THE PURGE
$       WRITE SYS$OUTPUT ""                             !  WAS NOT DONE
$       GOTO DO_DIR                                     !  AND BRANCH TO BUILD
$                                                       !  THE DIRECTORY FILE
$                                                       !
$ PRE_PURGE:                                            !  PURGE THE DISK:
$       WRITE SYS$OUTPUT ""
$       WRITE SYS$OUTPUT "Starting Purge ..."
$       PURGE'LOG' 'DISKDIR''ACCTMN''FILEMN'            !  PURGE THE DISK SO
$                                                       !  DUPLICATES WILL BE
$                                                       !  GONE, THIS WILL
$                                                       !  PREVENT PROBLEMS
$                                                       !  WITH THE GENERATION
$                                                       !  NUMBER
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< BUILD DIRECTORY FILE >>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ DO_DIR:                                               !  DIRECTORY:
$       DIR/COL=1/OUTPUT='DISK'.DIR/EXCLUDE=*.DIR 'DISKDIR''ACCTMN''FILEMN'
$                                                       !  BUILD A DIRECTORY
$                                                       !  FILE OF THE WHOLE
$                                                       !  DISK
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< OPEN DATA FILES >>>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ OPEN_FILES:                                           !  OPEN ALL NEEDED
$                                                       !  DISK FILES:
$       OPEN/READ  DIRFIL 'DISK'.DIR                    !  THE DIRECTORY FILE
$       OPEN/WRITE CMDFIL 'DISK'.COM                    !  THE COMMAND FILE FOR
$                                                       !  THE FILE COMPRESS
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< PROCESS DIRECTORY FILE >>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$       WRITE SYS$OUTPUT ""                             !  LET USER KNOW WHAT
$       WRITE SYS$OUTPUT "Building Command File ..."    !  IS GOING ON AT THIS
$       WRITE SYS$OUTPUT ""                             !  STEP
$                                                       !
$ READ_NAME:                                            !  READ PROGRAM NAMES:
$       READ/END=CMD_DONE DIRFIL FILNAM                 !  READ A RECORD FROM
$                                                       !  THE DIRECTORY FILE
$       FILNAM_LEN = 'F$LENGTH( FILNAM )'               !  GET THE LENGTH OF
$                                                       !  THE RECORD
$       IF FILNAM_LEN .EQ.0 THEN GOTO READ_NAME         !  IF A NULL RECORD
$                                                       !  THEN BRANCH TO
$                                                       !  READ THE NEXT RECORD
$       FILNAM_CHK = 'F$LOCATE( "Total of " , FILNAM )' !  SCAN THE RECORD FOR
$                                                       !  THE END OF A
$                                                       !  DIRECTORY STRING
$       IF FILNAM_CHK .LT. FILNAM_LEN THEN GOTO READ_NAME
$                                                       !  IF FOUND WE DO NOT
$                                                       !  WANT IT SO BRANCH TO
$                                                       !  READ THE NEXT RECORD
$       FILNAM_CHK = 'F$LOCATE( "Directory " , FILNAM )'
$                                                       !  SCAN THE RECORD FOR
$                                                       !  THE BEGINNING OF A
$                                                       !  DIRECTORY STRING
$       IF FILNAM_CHK .LT. FILNAM_LEN THEN GOTO PROC_ACCT
$                                                       !  IF FOUND THE BRANCH
$                                                       !  TO PULL THE ACCOUNT
$                                                       !  NAME FROM THIS
$                                                       !  RECORD
$                                                       !
$ PROC_NAME:                                            !  OTHERWISE WE HAVE A
$                                                       !  FILE NAME RECORD
$                                                       !  WHICH WE WANT TO
$                                                       !  COPY/DELETE/RENAME:
$       FILREO := 'ACCT''FILNAM'                        !  ADD THE ACCOUNT NAME
$                                                       !  TO THE FILE NAME
$       CPYCMD := $ COPY'LOG' 'FILREO' TMP.TMP          !  BUILD COMMAND STRING
$                                                       !  TO COPY FILE INTO
$                                                       !  A TEMPORARY FILE
$       WRITE CMDFIL CPYCMD                             !  WRITE THIS COMMAND
$                                                       !  STRING TO THE
$                                                       !  FILE COMPRESS
$                                                       !  COMMAND FILE
$       DELCMD := $ DELETE'LOG' 'FILREO'                !  BUILD ANOTHER
$                                                       !  COMMAND STRING TO
$                                                       !  DELETE THE ORIGINAL
$                                                       !  FILE
$       WRITE CMDFIL DELCMD                             !  WRITE THIS STRING
$                                                       !  TO THE COMMAND FILE
$       RNMCMD := $ RENAME'LOG' TMP.TMP 'FILREO'        !  BUILD THE COMMAND
$                                                       !  STRING TO RENAME THE
$                                                       !  TEMPORARY FILE AS
$                                                       !  THE ORIGINAL THUS
$                                                       !  PRESERVING THE
$                                                       !  GENERATION NUMBER
$       WRITE CMDFIL RNMCMD                             !  WRITE THIS COMMAND
$                                                       !  INTO THE COMPRESS
$                                                       !  COMMAND FILE
$       GOTO READ_NAME                                  !  BRANCH TO READ THE
$                                                       !  NEXT DIRECTORY
$                                                       !  RECORD
$                                                       !
$ PROC_ACCT:                                            !  PROCESS ACCOUNT
$                                                       !  NAME:
$       ACTSTG := 'FILNAM'                              !  COPY THE RECORD
$                                                       !  INTO ANOTHER
$                                                       !  VARIABLE FOR
$                                                       !  CLARITY OF WHAT
$                                                       !  IS BEING WORKED ON
$       ACTSTG_LEN = 'F$LENGTH( ACTSTG )'               !  GET ITS LENGTH
$       ACCT_CHK = 'F$LOCATE( " " , ACTSTG )'           !  SCAN THE RECORD
$                                                       !  FOR THE ACCOUNT
$                                                       !  NAME USING A SPACE
$                                                       !  AS THE DELIMITER
$       ACCT_CHK = ACCT_CHK + 1                         !  ADJUST TO GET THE
$                                                       !  OFFSET OF THE FIRST
$                                                       !  CHARACTER IN THE
$                                                       !  ACCOUNT NAME
$       ACCT_LEN = ACTSTG_LEN - ACCT_CHK                !  CALCULATE THE LENGTH
$                                                       !  OF THE ACCOUNT NAME
$       ACCT := 'F$EXTRACT( ACCT_CHK , ACCT_LEN , ACTSTG )
$                                                       !  PULL THE ACCOUNT
$                                                       !  NAME OUT OF THE
$                                                       !  DIRECTORY RECORD
$                                                       !  READ THE NEXT RECORD
$       GOTO READ_NAME                                  !  BRANCH TO READ THE
$                                                       !  NEXT DIRECTORY FILE
$                                                       !  RECORD
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<< VERIFY FILE COMPRESS >>>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ CMD_DONE:                                             !  COMMAND FILE BUILT:
$       CLOSE DIRFIL                                    !  CLOSE THE DIRECTORY
$                                                       !  FILE
$       DELETE/LOG 'DISK'.DIR;*                         !  DELETE THE
$                                                       !  DIRECTORY FILE
$       DELCMD := $ DELETE/LOG 'DISK'.COM;*             !  BUILD A STRING TO
$                                                       !  DELETE THE COMMAND
$                                                       !  FILE
$       WRITE CMDFIL DELCMD                             !  WRITE THIS STRING
$                                                       !  INTO THE COMMAND
$                                                       !  FILE
$       WRITE CMDFIL "$ EXIT"                           !  WRITE THE EXIT
$                                                       !  COMMAND TO THE
$                                                       !  COMMAND FILE
$       CLOSE CMDFIL                                    !  CLOSE THE FILE
$                                                       !  COMPRESS COMMAND
$                                                       !  FILE
$       WRITE SYS$OUTPUT ""
$       WRITE SYS$OUTPUT "Command File ", DISK, ".COM Ready"
$                                                       !  TELL THE USER THAT
$                                                       !  THE COMPRESS COMMAND
$                                                       !  FILE IS READY
$       INQUIRE VERIF "Compress Files ( Y/N=CR ) "      !  PROMPT OPERATOR FOR
$                                                       !  VERIFCATION TO
$                                                       !  PROCEED
$               IF VERIF THEN GOTO CMP_DO               !  ON Y BRANCH TO
$                                                       !  COMPRESS DISK FILES
$       WRITE SYS$OUTPUT "Command File Saved, File Compress Not Done"
$                                                       !  OTHERWISE NO SO
$                                                       !  DISPLAY A MESSAGE
$       GOTO DISK_IN                                    !  AND BRANCH TO GET
$                                                       !  ANOTHER LOGICAL
$                                                       !  DISK NAME
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<< EXECUTE FILE COMPRESS COMMAND FILE >>>>>>>>>>>>>>> &
$                                                       !
$ CMP_DO:                                               !  COMPRESS DISK FILES:
$       WRITE SYS$OUTPUT ""                             !  DISPLAY A MESSAGE TO
$       WRITE SYS$OUTPUT "Beginning File Compress ..."  !  LET THE USER KNOW
$                                                       !  WHAT IS GOING ON
$       @'DISK'.COM                                     !  EXECUTE THE
$                                                       !  FILE COMPRESS
$                                                       !  COMMAND FILE
$       WRITE SYS$OUTPUT ""                             !  TELL USER THAT IT
$       WRITE SYS$OUTPUT "Disk File Compress Completed"
$                                                       !  IS DONE
$       WRITE SYS$OUTPUT ""                             !  DISPLAY A MESSAGE
$       WRITE SYS$OUTPUT "A DSC-2 Disk Compress Is Recommended To"
$       WRITE SYS$OUTPUT "Prevent Disk Fragmentation"   !  TO REMIND THE
$       WRITE SYS$OUTPUT ""                             !  USER ABOUT THE
$                                                       !  DISK COMPRESS
$       GOTO DISK_IN                                    !  BRANCH TO GET
$                                                       !  ANOTHER LOGICAL
$                                                       !  DISK NAME
$                                                       !
$ !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< END UTILITY >>>>>>>>>>>>>>>>>>>>>>>>>>>>>> &
$                                                       !
$ CMP_END:                                              !  END OF PROCESS:
$       WRITE SYS$OUTPUT ""                             !  PRINT A BLANK LINE
$       WRITE SYS$OUTPUT "End Of Transcomm Disk File Compress Utility"
$                                                       !  PRINT A CLOSING
$                                                       !  MESSAGE
$       EXIT                                            !  EXIT THIS COMMAND
$                                                       !  FILE
$                                                       !
```

DEC 11/70 Users

Take the sure path to growth

With a BTI 8000 32 bit multiprocessor system

Now, for less than the cost of another 11/70, you can have a system that handles up to 200 on-line, interactive terminals, 8 CPUs, 16 Mbytes of main memory and 8 Gbytes of mass storage — the BTI 8000. And, you can bring your BASIC PLUS applications along, thanks to our CVT Translator.

The CVT Translator is an automated source language and file conversion system that takes the hard work out of the conversion process. So, converting your BASIC PLUS programs into BTI's more comprehensive BASIC/8000 is both fast and easy.

The BTI 8000's key to growth is Variable Resource Architecture (VRA), a pool of modular hardware resources with a single, self regulating operating system. With VRA you have almost unlimited flexibility in configuring your system.

Along with this, a virtual machine environment, a hierarchical account structure, and fail-soft architecture eliminate any worries about security, control or downtime. You also have device-independent programming, the ability to mix both batch and interactive operations, and simultaneous use of COBOL, FORTRAN, PASCAL, BTI BASIC, and BASIC/8000.

As for reliability and service, they're ensured. BTI has been using remote diagnostics for more than 10 years, and currently supports over 3500 computer systems worldwide.

So, if you're looking for growth in performance without an equivalent growth in budgets, migrate to the BTI 8000. For complete details, contact your nearest BTI office.

BTI COMPUTER SYSTEMS

Corporate Offices: 870 West Maude Avenue, Sunnyvale, CA (408) 733-1122; Regional Offices: Piscataway, NJ (201) 457-0600; Palatine, IL (312) 397-9190; Atlanta, GA (404) 396-1630; Sunnyvale, CA (408) 749-0500. In the United Kingdom: Birmingham (021)-477-3846. Sales Offices in major U.S. cities.

CIRCLE 136 ON READER CARD

FEED BACK

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

Some comments about the proposed EDT 2.0 standard initializer file:

This article is a response to a proposal for a standard startup command file for the DEC standard editor — EDT, which was published in this magazine by David Spencer of Infinity Software Corporation (March & April issues).

The proposed command file enhanced the capabilities of EDT and provided examples of the following:

1. Definition of additional keys.
2. Setting of terminal characteristics & entity delimiters.
3. Macros for toggling such definitions & settings and for executing command sequences.

I would like to suggest some changes and additions to the proposed standard, bearing in mind the following ideas:

1. As EDT runs on several operating systems and terminal types any extensions to it should be equally compatible. This precludes the use of, for instance, the CTRL/Y key sequence because of its abortive effect under VMS.
2. Since there are many actual and potential users of EDT who use non DEC terminals, some provision should be made for enabling the display of an alternative keypad diagram.
3. There should be no redefinition of preassigned keypad or other functions which in any way comprises the validity of information in DEC supplied manuals, or correspondingly impairs the normal operation of EDT. Thus the GOLD key synonyms for tab control operations (for e.g., GOLD T for CTRL/T etc.) should not be redefined.
4. Keystroke sequences should be arranged so that:
   a) Commonly used commands require fewer keystrokes than those less commonly used.
   b) Commands should have a mnemonic character wherever possible.
   c) Frequently used commands should not be similar or physically adjacent to commands whose action is not easily reversible, clearing a buffer for instance.
5. Finally room should be left for some degree of flexibility to allow limited, possibly application specific, customization of the startup command file while maintaining maximum compatibility with the standard.

The command file (Figure 1) is currently used by most programmers in this installation, on both VAX 11/780 and PDP 11/34 machines running under VMS and RSX-11M respectively. As such it reflects our requirements and I do not propose that it should be implemented as a universal standard, but taken perhaps as a useful starting point for other installations.

I shall outline the main differences from and additions to the previous proposal.

HELP

GOLD H will display the keypad diagram shown in Figure 2 and may then be used to return to the text being edited.

BUFFER COMMANDS

All buffer commands begin with the GOLD key and most are comprised of one other character which is alphabetic and chosen for mnemonic or alliterative quality; e.g., C, K, P, and R for cut, copy, paste and replace respectively. The help diagram referred to above is in fact just a buffer containing the diagram which is loaded from an input file EDTINI.ADM.

Additional commands are the options to delete to the beginning and end of the current buffer using GOLD CTRL/↑ and GOLD $ respectively. Also included is the facility for copying both individual lines and selected text ranges to the end of the paste buffer. This is useful for collecting source code from one program for use in another. GOLD DELETE (rubout key) can be used to clear a specified buffer. GOLD W the 'backup' command, suggested by its equivalent in the SOS editor, causes the main buffer to be written to an output file called EDTFIL.BAK. On the VAX the /RECOVER facility will restore edits after a CTRL/Y interrupt, but not, in our experience, after a system crash as the journal file is itself corrupted, hence the utility of the command.

GENERAL COMMANDS

The 1st 8 of these consist of 4 pairs with complementary functions. CTRL/F and CTRL/B scroll the display forward and back by 20 lines, or multiples thereof, if a repeat count is specified. A repeat count may be specified for all definitions listed in parentheses. GOLD G and GOLD : invert the case of the next word and the next word excluding the 1st letter respectively. GOLD . and GOLD F insert and find a mark comprised of the character sequence #&#. The final pair of commands include arguably the most useful of all:

CTRL/V copies a word from the line above, and
CTRL/R copies a letter from the line above.

CTRL/V saves time and effort in 2 circumstances:
Consider the following: (cursor = _)

```
IF condition THEN BEGIN;
  CALL TOSCREEN (' Message', other parameters . . .
  _                          CTRL/V takes care of any indenting
  CALL TOSCREEN (' _         CTRL/ and V 3 times copies this far.
```

CTRL/R may be used to copy character sequences, with non identical characters being entered individually. CTRL/V works best with the default word delimiters.

The remaining 8 general commands may be used as follows:

GOLD J

For justifying text:
This is a line of text and the right hand margin is right here:
This is a line of text without an aligned right hand margin

To align the right hand margin when a line is 'n' spaces shorter than its predecessor: type CTRL/H (or backspace) followed by GOLD 'n' GOLD J and the words will be double spaced and the margins aligned. On occasion it will be preferable to plod backwards and forwards justifying by eye, additional spaces are less noticeable between longer words.

GOLD @

Serves as a global substitution command. A query option or alternative command incorporating it is redundant since this can be achieved using the existing keypad functions.

GOLD |

Is useful for drawing vertical lines, histogram bars etc. GOLD 20 GOLD | with '| |' in the paste buffer will give a vertical bar 20 characters high.

GOLD N

Moves the current line to the top of the screen, seems to work in the main buffer only.

CTRL/N

May also be used to move the current line. This command swaps the current line with the next, depending on the direction set. With direction set forward GOLD CTRL/N can be combined with CTRL/H (or backspace) to reorder a list without the 'delete line, move, undelete line' rigamarole.

GOLD CTRL/P and GOLD CTRL/F

Move to the next and fill selected paragraphs respectively, functionally as described by David Spencer.

EDITOR OP COMMANDS

This set of commands comprises 4 pairs. Their functions are self explanatory and I will not delay with them, except to say that I have not chosen the keyboard symbols used entirely arbitrarily.

TERMINATING COMMANDS

Consist of two pairs. Originally I used GOLD Q to quit; however, since there was a possibility of issuing this command accidentally when intending to make a backup copy of the current buffer (by typing GOLD W) I changed to GOLD ? — which is alliterative at least, and not likely to be entered accidentally. GOLD Z is synonomous with CTRL/Z followed by EXIT (or GOLD COMMAND followed by EXIT) and causes a normal exit, deleting the journal file. The latter may be saved if /SAVE is appended to the terminating command. GOLD # and GOLD + serve as EXIT/SAVE and QUIT/SAVE respectively. Finally,

SPECIAL COMMANDS

The examples included in the command file are illustrative of some programming language specific function definitions. All the languages mentioned are used in Petroconsultants (and ADA will probably be added to the list at some future date!).

COBOL

Those trained in structured programming techniques using data driven design (a la Jackson) will appreciate this one. The convention whereby every paragraph is performed through a dummy exit paragraph leads to typing tedium. Now you can enter the paragraph name and type GOLD \ and end up with:

```
PERFORM 'PARAGRAPH-NAME' THRU 'PARAGRAPH-NAME'-X
```

and the cursor sits at the end of the line waiting for a full stop or a comma.

PL/I

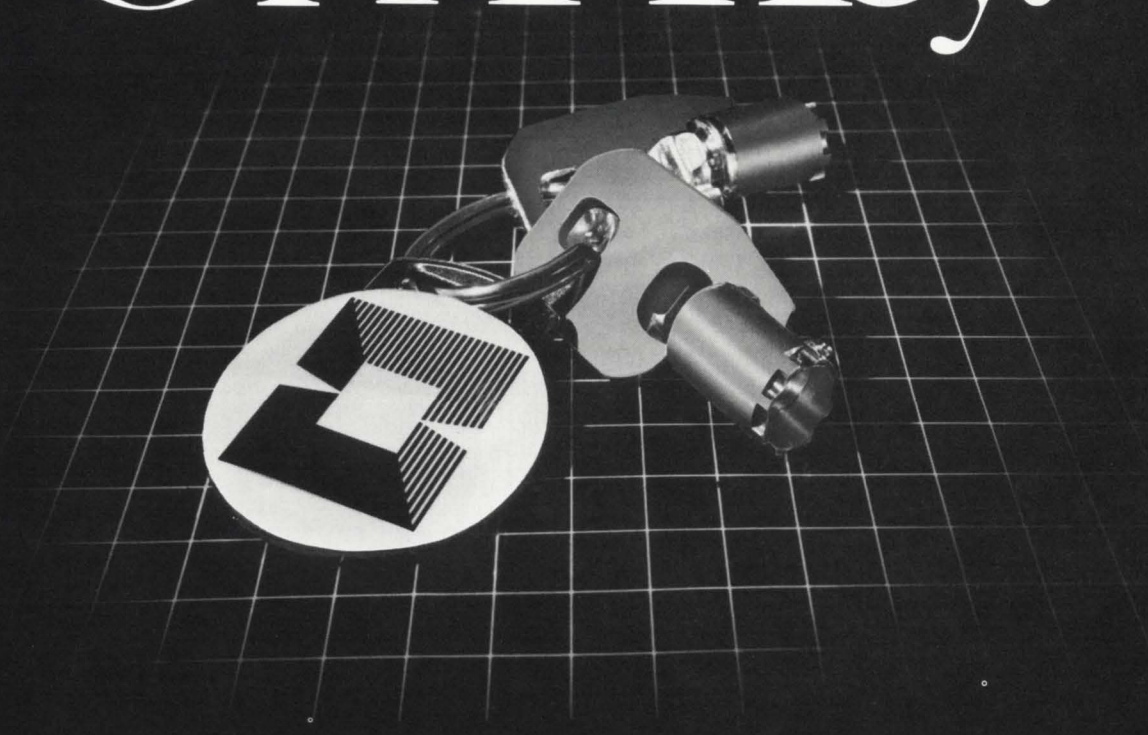
Typing comment delimiters can be a bit of a chore, especially on terminals with a shifted * . CTRL/P does the job and positions the cursor (__).

```
... FIXED BINARY (15); /* __ */
```

FORTRAN

GOLD __ will insert a ruler above the current line (provided one is not at the very end of the line). The 72nd

VAX Relational Database At The Turn Of A Key.



Unlock The Full Power Of Your VAX* With Relational Database Management Systems From Britton-Lee.

SYSTEM 300/600

Visit us at:
NCC '82
June 7–10
Booth #5712

CIRCLE 141 ON READER CARD

Unshackle Your VAX.

Your VAX will perform more powerfully and more responsively when you release it from time-consuming database management chores. Britton-Lee's SYSTEM 300 and SYSTEM 600 are your keys to relational databases of up to 11 billion bytes.

Hardware Solution.

The specialized, unique database hardware and software of Britton-Lee's SYSTEM 300 and SYSTEM 600 are installed by Britton-Lee and fully supported through training and follow-on maintenance. Your VAX will be more productive than you ever imagined it could be.

User Friendly.

Special software includes IDL Query Language, high-level interface to VAX-11 FORTRAN and COBOL, and DBA Utilities. The SYSTEM 300 uses the DEC UNIBUS™ interface for easy system integration; it also has 1 megabyte memory and controls up to 4 SMD disk drives for a total database capacity of almost 3 billion bytes. Up to 6 megabytes of memory are included with the SYSTEM 600, which can manage almost 11 billion bytes of database.

Relational Ease Of Use.

Both the SYSTEM 300 and SYSTEM 600 are fully relational database management systems. They make it easy for relatively unskilled users to create, modify, and access the database without the expense of extensive programming support.

Get the SYSTEM 300/600 facts now! Write or call Britton-Lee to learn how to turn the key to full power for your VAX.

*VAX, VMS, DEC UNIBUS are registered trademarks of Digital Equipment Corporation.

Britton Lee, Inc.

90 Albright Way
Los Gatos, CA 95030
(408) 378-7000 Telex 172-585

Get the VAX facts! Fill out this coupon and drop it in the mail now.

- ☐ SYSTEM 300/600 Information
- ☐ DBMS Seminar Schedule
- ☐ Please have a Britton-Lee sales representative call.

90 Albright Way
Los Gatos, CA 95030
Attn: Peter Moulds

NAME _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

PHONE #( ) _____ SN _____

Your Computer Configuration _____

_____

character is marked with a 'V'. Whereas setting screen wrap to 72 will not necessarily tell one when one is approaching the end of a line, the ruler may be readily used for positional reference. It may be carried forward using CTRL/H and CTRL/N as described above, and may of course be used in other circumstances.

BP2

I have changed my key definitions to accord with those of David Spencer as they are better. Incidentally, the indentation of DEF K in the listing where it occurs is to prevent overprinting as the command definitions contain an embedded carriage return. To recap, these commands toggle the definition of carriage return between its normal value and one whereby EDT prefixes that with '&' for line continuation.

Conclusion:

I do not think that a completely standard initializing command file is desirable or practicable. However, the proposal has been thought provoking and worthwhile. With the enhancement in the EDTINI.EDT file EDT is quite comparable to the kernel of some good word processing packages, and I look forward to the day when DEC produces an integrated programmable editor cum document processing system.

Finally, just an advisory note on the comparative effects of different editors on system performance; on the VAX, and measuring in the somewhat subjective units of 'space invader players', I rate them as follows.

TECO	c1.5 SPIVS
EDT	c0.5 SPIV
SOS	c0.2 SPIV
EDI	c0.1 SPIV


```
!                    ************************
!                    * EDT 2.0 STARTUP FILE *
!                    ************************
!
INCLUDE DRU:[PAULUS.SUB]EDTINI.ADM =EDTADM
!
!                    MACROS
!
DEFINE MACRO APP
INSERT=APP 10;CO SELECT TO=APPENDER
INSERT=APP 20;FIND=PASTE
INSERT=APP 30;MOVE =APPENDER TO END
INSERT=APP 40;FIND=MAIN
!
DEFINE MACRO ADM
INSERT=ADM 10;FIND=EDTADM
INSERT=ADM 20;DEF K GOLD H AS "EXT BYE."
!
DEFINE MACRO BYE
INSERT=BYE 10;FIND LAST
INSERT=BYE 20;DEF K GOLD H AS "EXT ADM."
FIND=MAIN.
!
!                    SPECIAL
!
!                                        PERFORM: COBOL
!
DEF K GOLD \ AS "(-W DW IPERFORM  THRU -X -2W -C UNDW 2W UNDW EL)."
!
!                                        RULER: FORTRAN
!
 BL V)."   DEF K GOLD _ AS "(+C BL TD IC...:....1....:....2....:....3....:....4....:....5....:....6....:....7.V..:....
!
!                                        COMMENT: PL/I
!
DEF K CONT P AS "(I/*   */ -4C)."
!
!                                        AMPERSAND: BP2
!
.'."  DEF K GOLD & AS "EXT DEF K CONT M AS 'I &
.'."  DEF K GOLD CONT M AS "EXT DEF K CONT M AS 'I
!
!                    BUFFERS
!
DEF K GOLD CONT A AS "EXT APP."
DEF K GOLD CONT L AS "(ADV D+NL UNDL SEL L APPENDSR UNDL L)."
DEF K GOLD B AS "EXT FIND=?'Move to buffer: '.."
DEF K GOLD C AS "CUTSR=?'Cut to Buffer: '.."
DEF K GOLD H AS "EXT ADM."
DEF K GOLD I AS "EXT INCLUDE ?'Input file: ' =?' Buffer: '."
DEF K GOLD K AS "EXT COPY SELECT TO=?'Copy to buffer: ' ; F L."
DEF K GOLD M AS "EXT FIND=MAIN.."
DEF K GOLD O AS "EXT WRITE ?'Output file: ' =?' From buffer: '."
DEF K GOLD P AS "PASTE=?'Put buffer: '."
DEF K GOLD R AS "CUTSR=DELETE PASTE=?'Replace buffer: '."
DEF K GOLD S AS "EXT SHOW BUFFER."
DEF K GOLD W AS "EXT WRITE EDTFIL.BAK."
DEF K GOLD DELETE AS "EXT CLEAR ?'Clear buffer: '."
DEF K 101 AS "EXT DELETE BEFORE."
!=DEF K GOLD CONT ^ AS "EXT DELETE BEFORE."
DEF K GOLD $ AS "EXT DELETE REST."
DEF K GOLD L AS "EXT FIND LAST."
!
!                    GENERAL
!
DEF K GOLD F AS "S$#&#$$."
DEF K CONT B AS "(-20V)."
DEF K GOLD G AS "(ADV SEL W CHGCSR)."
DEF K GOLD : AS "(ADV C SEL W CHGCSR)."
DEF K CONT F AS "(+20V)."
DEF K GOLD . AS "I#&#-3C."
DEF K CONT V AS "(ADV -V DW UNDW V UNDW W)."
DEF K CONT R AS "(ADV -V DC UNDC V UNDC C)."
!
DEF K GOLD N AS "TOP."
DEF K GOLD V AS "(-C D-C C UNDC)."
DEF K CONT N AS "(D+NL L UNDL)."
DEF K GOLD CONT P AS "PAR."
DEF K GOLD CONT F AS "(SEL PAR FILLSR)."
DEF K GOLD ; AS "(ELPASTE)."
DEF K GOLD J AS "(ADV +WI )."
DEF K GOLD @ AS "EXT S/?'Change all '/?' to '/WHOLE."
!
!                    EDITOR OPERATING CHARACTERISTICS
!
!        (SEARCH)         {BOUND}          [TRUNCATE]       <SCREEN>
!
DEF K GOLD ( AS "EXT SET SEARCH GENERAL."
DEF K GOLD ) AS "EXT SET SEARCH EXACT."
DEF K GOLD } AS "EXT SET SEARCH BOUNDED."
DEF K GOLD { AS "EXT SET SEARCH UNBOUNDED.
DEF K GOLD ] AS "EXT SET TRUNCATE."
DEF K GOLD [ AS "EXT SET NOTRUNCATE."
DEF K GOLD > AS "EXT SET SCREEN 132."
DEF K GOLD < AS "EXT SET SCREEN 80."
!
!                    EXITS
!
DEF K GOLD Z AS "EXT EX."
DEF K GOLD ? AS "EXT QUIT."
DEF K GOLD # AS "EXT EXIT/SAVE."
DEF K GOLD + AS "EXT QUIT/SAVE."
!
!                    SET TERMINAL CHARACTERISTICS
!
SET MODE CHANGE
SET NOTRUNCATE
!
!                    SOME KEY SEQUENCES AVAILABLE FOR REDEFINITION
!
!DEF K GOLD Q AS ""
!DEF K GOLD CONT B AS ""
!DEF K GOLD CONT C AS ""
!DEF K GOLD CONT D AS ""
!DEF K GOLD CONT E AS "" etc.
```

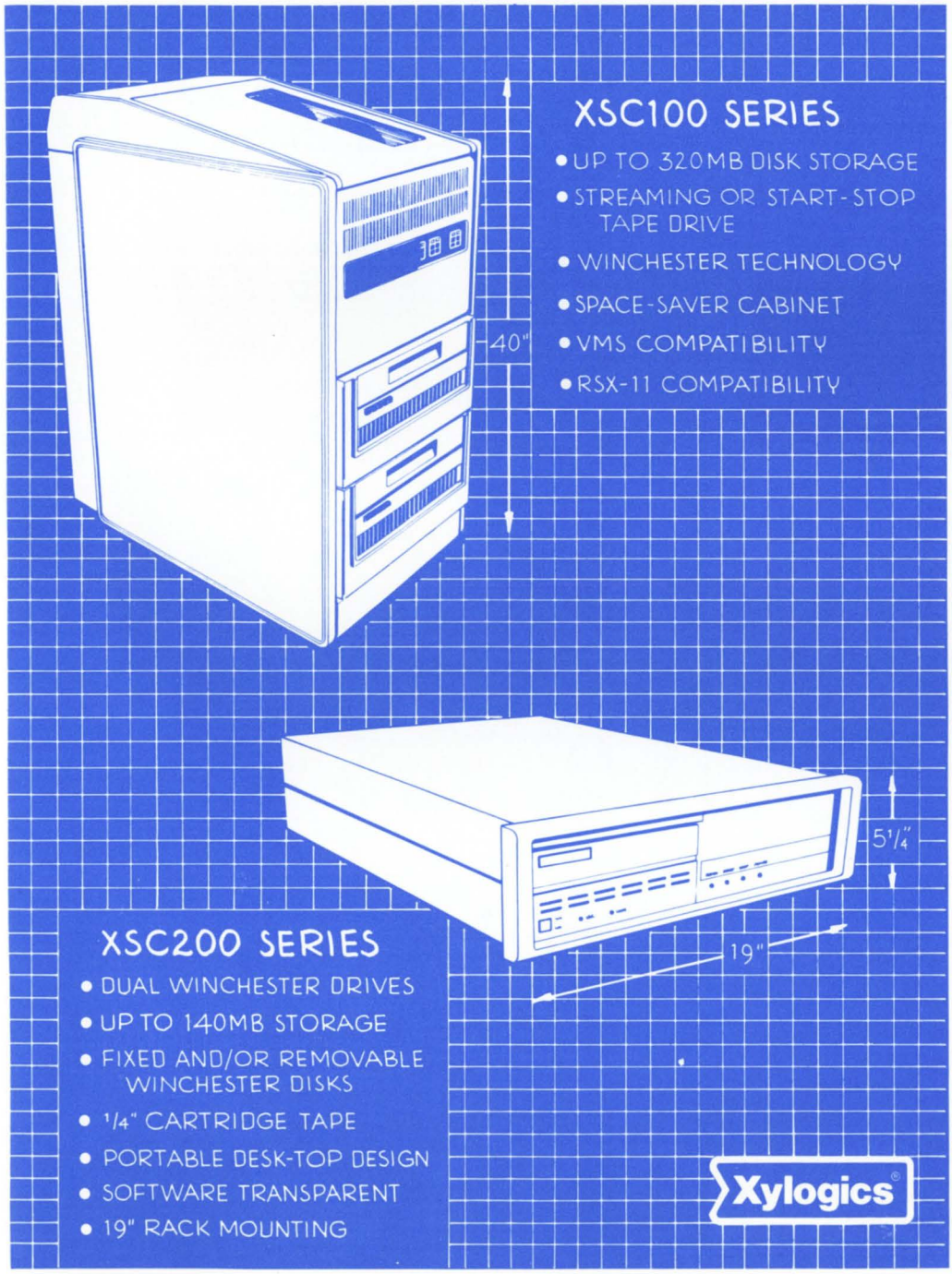
FIGURE 1. EDTINI.EDT COMMAND FILE

```
                                            +---------+---------+---------+   EDT keypad for
   RUB     Delete character                 !         !         ! DEL L   !    Lear Siegler
   LF      Delete last word                 !  GOLD   !  HELP   !         !      ADM 31
   CTRL/H  Move to start line               !         !         ! UND L   !
                                            +---------+---------+---------+---------+
   CTRL/A  Compute  tab level                         !  PAGE   ! FNDNXT  ! DEL W   !
   CTRL/D  Decrease tab level                         !    7    !    8    !    9    !
   CTRL/E  Increase tab level                         ! COMMAND !  FIND   ! UND W   !
                                                      +---------+---------+---------+
   CTRL/F  Fill text                                  ! ADVANCE ! BACKUP  ! DEL C   !
   CTRL/K  Define key                                 !    4    !    5    !    6    !
   CTRL/T  Adjust tabs                                ! BOTTOM  !  TOP    ! UND C   !
   CTRL/Z  Return to line mode                        +---------+---------+---------+---------+
                                                      ! WORD    !  EOL    !  CUT    !    E    !
** GOLD H to resume **                                !    1    !    2    !    3    !    N    !
                                                      !CHNGCASE ! DEL EOL ! PASTE   !    T    !
                                                      +---------+---------+---------+    E    !
+---------+---------+---------+---------+                       !  LINE   ! SELECT  !    R    !
! DOWN    !   UP    !  LEFT   !  RIGHT  !                       !    0    !    .    !         !
!   v     !   ^     !  <--    !   -->   !                       ! OPEN L  ! RESET   !  SUBS   !
! SECT    ! REPLACE ! APPEND  ! SPECINS !                       +---------+---------+---------+
+---------+---------+---------+---------+
```

FIGURE 2. EDTINI.ADM FILE, HELP BUFFER.

Storage Cells from Xylogics

More Data Storage for Your Money



Xylogics Storage Cells: A totally new way to solve data storage problems on VAX*, PDP-11*, LSI-11*, Nova/Eclipse** and Multibus*** compatible systems. We start with the proven performance, reliability and economy of our Peripheral Processors for 100% software compatible control. Then we add state-of-the-art peripherals like high-performance fixed disk and removable cartridge Winchesters, ¼" cartridge tapes, and reel-to-reel tapes for start-stop or streaming operation. And we wrap everything up in a series of modular cabinets that integrate perfectly with your computer packaging.

Storage Cells: They give you more choices, more capacity and more convenience in less space, and at lower cost than any other data storage subsystem on the market.

MORE SOLUTIONS FOR YOUR SYSTEM

Now you can forget about data storage limitations. With Storage Cells you can get the perfect combination of capacity and application-oriented features in an incredibly compact package. Consider the XSC100 for VAX and PDP-11 systems. In a single bay cabinet only 40" high, this Storage Cell gives you up to two 160MB disks plus a reel-to-reel streaming tape with emulation of DEC's RK07's or RM02's and TS04 peripherals.

The XSC200 for Q-bus,* Unibus,* Multibus, or Data General I/O bus systems is equally versatile. In one 5¼ inch high cabinet you can combine fixed and removable Winchester disks. Or you can have two 34MB or up to two 70MB Winchesters in the same 5¼" space. For Q-bus only, the XSC200 features up to 70MB of Winchester storage plus a 17MB cartridge tape drive.

Storage Cells: Innovative, cost-effective, state-of-the-art. Just what you'd expect from Xylogics. Call today for complete details. Boston (617) 272-8140, Chicago (312) 660-1460, New York (201) 691-2800, London (0753) 78921, San Francisco (408) 995-5205, Los Angeles (714) 966-0888, or headquarters (800) 225-3317.

U.S. Headquarters: 144 Middlesex Turnpike, Burlington, MA 01803 Tel: (617) 272-8140 (TWX 710-332-0262) European Headquarters: 46-48 High Street, Slough, Berks SL1 1ES U.K. Tel: (0753) 78921; Telex 847978

GOLD		CTRL	
Compute tab level *	A	Compute tab level *	
Move to named buffer	B	Scroll back 20 lines	
Cut to named buffer	C	Abort operation *	
Decrease tab level *	D	Decrease tab level *	
Increase tab level *	E	Increase tab level *	
Find and delete mark	F	Scroll forward 20 lines	
Change case of next word	G		
Move to/from Help buffer	H	Move to start of line *	
Load file into named buffer	I	Tab *	
Next word, insert space	J	Delete word before cursor *	
Copy to named buffer	K	Define key *	
Find last position	L	Insert formfeed *	
Move to main buffer	M	Insert carriage return *	
Move line to top of screen	N	Swop lines	
Write named buffer to file	O		
Paste to named buffer	P	Insert PL/I comment delimiters	
	Q	Continue *	
Replace with named buffer	R	Copy character from line above	
Show buffers	S	Stop *	
Adjust tab setting *	T	Adjust tab setting *	
	U	Delete to start of line *	
Transpose 2 preceding chars	V	Copy word from line above	
Write backup file	W	Refresh screen, eg after system msg *	
	X	Delete to start of line (VMS) *	
	Y	Abort edit saving journal file (VMS) *	
Exit	Z	Return to line mode *	
Insert mark	.	GOLD CTRL	
Change word case exc 1st ltr	:	---	
Delete to end of file	$	Append to paste buffer	A
End of line, paste	;		
Show ruler		Set mark, move paragraph, fill	F
Exit/save	#		
Quit	?	Append line to paste buffer	L
Quit/save	+		
Set BP2 ampersand return	&	Reset <cr> after BP2	M
Cobol perform thru	\		
Global substitute	@	Move to next paragraph	P
Set screen 132 characters	>		
Set screen 80 characters	<	Delete to top of file	^
Set notruncate	[		
Set truncate	]		
Set search general	(	Note:	
Set search exact	)	-----	
Set search unbounded	{	Original definitions	*
Set search bounded	}		
Clear named buffer	RUB/DELETE		

FIGURE 3. SUMMARY OF EDT FUNCTIONS.

LETTERS to the RSTS Pro . . .

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

CCLMAN — . . . continued from page 28

```
19080   IF ERL=1020 AND ERR=154% THEN                                   &
                SLEEP 1%                                                &
\               RESUME 1020                                             &

19090   IF ERL=2020 AND ERR=154% THEN                                   &
                SLEEP 1%                                                &
\               RESUME 2020                                             &

19100   IF ERL=10010 AND ERR=154% THEN                                  &
                SLEEP 1%                                                &
\               RESUME 10010                                            &

19110   IF ERL=10015 AND ERR=154% THEN                                  &
                SLEEP 1%                                                &
\               RESUME 10015                                            &

19900   PRINT FNERROR$("?Fatal error "+ERT$(ERR)+" at line "+           &
                NUM1$(ERL)+"- ABORTING")                                &
\       CLOSE #1%,#2%                                                   &
\       RESUME 32700                                                    &
20000   CHR6$       =CHR$(6%)                                           &
\       PRIV.ON$    =CHR6$+CHR$(-21%)+CHR$(0%)                          &
\       PRIV.OFF$   =CHR6$+CHR$(-21%)+CHR$(255%)                        &
\       PRIV.NEVER$=CHR6$+CHR$(-21%)                                    &
\       BELL$       =CHR$(7%)                                           &
\       DEBUG%      = 0%                    !NO DEBUG REQUESTED         &
\       ID.STG$     ="CCLMAN"                                           &
\       VERSION$    ="07"                                               &
\       VER.EDT$    ="07"                                               &
\       INSTAL$     = ERT$(0%)                                          &
\       OPEN.SPEC$ =FNPKG.LOC$                                          &
\ PRINT "OPEN CCL.DAT FILE SPEC = '";OPEN.SPEC$;"'"  IF DEBUG%          &
\       RETURN                                                          &

20100   DEF FNERROR$(ARG$)="?"+ID.STG$+" - "+BELL$+ARG$                 &

20200   DEF FNPKG.LOC$                                                                  &
!                                                                                       &
!       Function FNPKG.LOC      This function returns the PPN of the last               &
!                               opened file, usually the first program in               &
!                               a package of programs.  This information                &
!                               is returned in FSPEC format with a device               &
!                               suffixed if the device is not public.                   &
!                                                                                       &
!                                                                                       &
\       DIM SYS.CHNL.OPEN$(30%)                                                         &
\       CHANGE SYS(CHR$(12%)) TO SYS.CHNL.OPEN$                                         &
\       PKG.LOC$ = "("+NUM1$(SYS.CHNL.OPEN$(6%))+","+                                   &
                   NUM1$(SYS.CHNL.OPEN$(5%))+")"                                        &
\       PKG.LOC$ = "_"+CHR$(SYS.CHNL.OPEN$(23%))+                                       &
                   CHR$(SYS.CHNL.OPEN$(24%))+                                           &
                   NUM1$(SYS.CHNL.OPEN$(25%))+":"+PKG.LOC$                              &
                         IF SYS.CHNL.OPEN$(26%) AND 1%                                  &

20201   PKG.PROJ$   = SYS.CHNL.OPEN$(6%)                                                &
\       PKG.PROG$   = SYS.CHNL.OPEN$(5%)                                                &
\       FNPKG.LOC$ = PKG.LOC$                                                           &
\       FNEND

20300   !BREAK COMMAND                                                  &
        ERROR$=0%                                                       &
\       WS.KEY$=""                                                      &
\       WS.APP$=""                                                      &
\       WS.CMD$=""                                                      &
\       C% =INSTR(1%,CMD.VAL$,"-")      !KEY TERMINATOR                 &
\       GOTO 20320 IF C%=0%             !NO -, MAYBE =                  &
\       WS.KEY$=LEFT(CMD.VAL$,C%-1%)                                    &
\       CMD.VAL$=RIGHT(CMD.VAL$,C%+1%)                                  &

20320   C%=INSTR(1%,CMD.VAL$,"=")                                       &
\       GOTO 20350 IF C%                !MUST HAVE THIS                 &
\       PRINT FNERROR$("Missing '=' in argument")                       &
\       ERROR%=-1%                                                      &
\       RETURN                                                          &

20350   WS.CMD$=RIGHT(CMD.VAL$,C%+1%)                                   &
\       WS.TMP$=LEFT(CMD.VAL$,C%-1%)                                    &
\       IF WS.KEY$="" THEN                                              &
                WS.KEY$=WS.TMP$                                         &
        ELSE                                                            &
                WS.APP$=WS.TMP$                                         &

20370   GOTO 20384 IF LEN(WS.APP$)+LEN(WS.KEY$)>95%                     &
\       RETURN  IF REMOVE%                                              &
\       C2%=INSTR(1%,WS.CMD$,";")                                       &
\       GOTO 20390 IF C2%=0%            !MUST HAVE ;#                   &
\       WS.FILE$=LEFT(WS.CMD$,C2%-1%)                                   &
\       WS.LN$=RIGHT(WS.CMD$,C2%+1%)    !HOLDS 'PRIVxxxxxx'             &
\       WS.PRIV%=0%                                                     &
\       C3%=INSTR(1%,WS.LN$,"PRIV")                                     &
\       GOTO 20380 IF C3%=0%                                            &
\       WS.PRIV$ =LEFT (WS.LN$,4%)      !ISOLATE PRIV BIT               &
\       WS.PRIV% = -1% IF LEFT(WS.PRIV$,4%)="PRIV"                      &
\       WS.LN$   =RIGHT(WS.LN$,5%)      !GET LINE NUMBERS               &

20380   WS.LN$=CVT$$(WS.LN$,2%)         !MAKE SURE OK                   &
\       WS.LINE%=VAL(WS.LN$) !BREAK UP LINE NUMBER                      &
\       CHANGE SYS(CHR$(6%)+CHR$(-10%)+WS.FILE$) TO M%                  &
\       S%=M%(29%)+SWAP%(M%(30%))       !FLAG WORD                      &
\       IF (S% AND 8192%)<>0% AND (S% AND 16384%)<>0%                   &
                AND (S%>=0%)            !NO TRANS OF LOGICAL            &
                THEN WS.FILE$="_"+WS.FILE$                              &
\                    GOTO 20380                                         &
20382   WS.FILE$=""                                                     &
\       GOTO 20395 IF (S% AND 1%)=0% OR                                 &
                      (S% AND 2%)<>0% OR                                &
                      (S% AND 4%)<>0% OR                                &
                      (S% AND 256%)<>0% OR                              &
                      (S% AND 512%)<>0%                                 &
\       WS.FILE$=RAD$(M%(7%)+SWAP%(M%(8%)))+                           &
                RAD$(M%(9%)+SWAP%(M%(10%)))     !FILENAME               &
\       WS.FILE$=WS.FILE$+"."+                                          &
                RAD$(M%(11%)+SWAP%(M%(12%)))                            &
                      IF (S% AND 16%)<>0%       !EXT SPECIFIED          &
\       WS.FILE$="["+NUM1$(M%(6%))+","+NUM1$(M%(5%))+"]"+WS.FILE$       &
```

THE RSTS CRYSTAL BALL — Part 1

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

This is the first in a series of articles on new RSTS/E updates, undocumented features, and bugs. Most of the more active RSTS/E users hungrily await new releases from DEC. Often these users are rather disappointed at what they see, or don't see, in new versions of RSTS. Many people are concerned about the future directions of RSTS. In this column, I hope to present information which will be of interest to all of these users.

All of the material contained in this column is based on short talks with the RSTS developers, peeks at past and present RSTS sources, a solid knowledge of RSTS internals, and partially on the opinion of myself and others not necessarily associated with DEC. The information presented here is believed to be an accurate picture of the directions in which RSTS is heading, however DEC is under no commitment to support their product in the manner in which I describe it. Keeping these facts in mind, I welcome you to a look into the future ...

While I will try to make this column intelligible to as broad a range of RSTS users as possible, I do not wish to rewrite the book on RSTS system concepts. I intend to present information which is fairly technical in nature, and therefore I expect the reader to have a reasonable understanding of RSTS monitor operations, structures, etc. Also, the reader will find familiarity with MACRO-11 and the PDP-11 instruction set useful.

GENERAL

I am sure the questions that most people are asking currently are about the latest RSTS release, version 7.1. What has changed since 7.0? Internally, quite a number of things, although most of these will not affect the average user.

DEC has done next to nothing to solve the problem of RSTS security (or insecurity, as the case may be). DEC is aware of the problem, but it is highly doubtful that they will do anything about it in the near future. Users are going to have to rely on in-house software, or, better, one of the available security packages. Several such packages exist, however you must know what you are buying. Some are nothing more than patches to existing DEC software. Others, if improperly installed (which is EASY to do) will cause far more security holes than they close up. The wise choice would be to go with something which replaces existing DEC software, and is not written in BASIC-PLUS.

INTERNAL SYSTEM STARTUP CHANGES

When the START (or line-feed) option of INIT is executed to startup RSTS, INIT prints its various prompts and informational messages and builds a "jam" table for the monitor. This is a table of information which is to be "jammed into" the monitor once it is loaded into memory. INIT also makes hundreds of checks of the hardware configuration, system default run-time system, swap files, etc. Finally, INIT moves one or more loading routines to various "safe" places and jumps into them to load the RSTS monitor. Once RSTS gains control, it initializes several minor things (such as the maximum job size for the "null" run-time system, which is set to current SWAP MAX) and forces the terminal service to create a job on KB0:. Under version 7.0, the monitor completes its startup by putting the newly created job in a FIP wait, and dispatching to the login code (LIN). LIN notices that the system disk is not mounted, logs the job into the system library account (normally [1,2]), and then goes and dispatches to mount (MNT) in order to mount the system disk. Under 7.1, the monitor puts the job in a FIP wait, but dispatches to an internal FIP function called STA (for START, naturally). This function calls LIN and then MNT to log in the first job and mount the system disk, and also loads and sets up overlay sections of the monitor which are supposed to be resident.

Overall, the startup code for 7.1 is cleaner, however it is much more complex due to the selective overlay loading, and the new FIP buffer pool scheme. It has been suggested that it is theoretically possible to patch the monitor to make modules resident or non-resident after the SIL has been linked. This has not been tested, and depends on whether or not SILUS is doing some calculations for INIT, or if INIT is also doing these calculations. If the latter is true, it is possible that a module residency table in the monitor could be changed at will and, upon re-booting the SIL, change the modules which are memory resident.

One rather interesting note: Try sitting on control/T while bringing up RSTS, just after INIT(.SYS) finishes any final initialization. You will probably be able to catch your RSTS job in a startup wait, i.e. FP(STA).

TERMINAL SERVICE

Several minor changes were made to the terminal service between 7.0 and 7.1, including support for FMS V1.5, two new terminal features (GAG and BREAK), and multiple private delimiters, all of which were fairly trivial to implement. I can't say much for the new terminal "features", the first of which is a fix for a long-standing oversight, and the second which removes a supposed feature which has always been far more annoying than useful.

Don't Buy Another VT-100* Until You Compare It To Our "SMALL WONDER"

PRODUCT FEATURES	VT-100	SW10
Printer Port	Opt	STD
Green or White Monitor	Opt	STD
20 MA Current Loop	Opt	STD
Non-Glare Monitor	Opt	STD
Programmable Function Keys	No	STD
Small Footprint for Desk Top Use	No	STD
English Language Set-Up Mode	No	STD
Graphic Character Set	Std	Std
Smooth Scroll	Std	Std
Split Screen	Std	Std
24 Lines/132 Column Characters	OPT	No
Blink & Underline Visual Attributes	OPT	No
Composite Video	STD	No
Double High, Double Wide Characters	STD	No

Standard Unit List Price . . . $1675.

$899.

Available Now!

See Us At
NCC Booth A-419

Call For Your FREE 30 Day Trial Unit Today!

1-800-854-6781

For further information call or send this coupon to:

GTC

General Terminal Corporation
14831 Franklin Avenue, Tustin, CA 92680
(714) 730-0123

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone (____) _____

(GTC is currently seeking Independent Sales Organizations. Inquiries welcomed.)

*Registered trade mark of Digital Equipment Corporation

CIRCLE 128 ON READER CARD

A SHORTAGE OF SMALL BUFFERS

By Tom Britton, CBL Canterbury Ltd,
Box 13147, Armagh St,
Christchurch, New Zealand

Small buffers are known to be a problem on "large" RSTS systems. The following situation illustrates a side to the small buffer problem that I didn't anticipate.

CBL is, among other things, a timesharing service bureau with a large number of 11/70's running RSTS. One of our clients has a full 11/70 to itself, and normally operates 35 to 40 terminals simultaneously. Small buffers are a severe problem on this client's 11/70; normally there are 50 to 70 free, occasionally dipping below 40. RSTS on their machine is built to obtain as many small buffers as possible (e.g., no statistics), and is run with as few detached jobs as possible (ERRCPY, OPSER and QUEMAN only, during heaviest load periods). Naturally they have a large XBUF for directory and data cacheing. Their machine is configured with 3 DH's, 2 RP06's, FPU, a TE16, and 1 Mb MOS memory.

Recently, their private disk, which contains the swapfiles, one very large data file, and miscellaneous other files, was rebuilt. For a variety of reasons, it was re-built with NO optimization. The swapfiles ended up at the outside edge; few files had clustersizes greater than 8; directories were built as needed; etc.

The result of this reorganization, was disaster in terms of system performance.

We appeared to lose something like 20-40 small buffers. The maximum number of jobs we could run simultaneously was reduced by 3 or 4; we ran out of small buffers very frequently ("no buffers" messages), and chronically operated at or below the magic 40 limit ("no logins").

The disk was re-built a second time; this time optimizing everything (using the DSU utility of Software Techniques' DSKIT). Now we're back to "normal". Still with fewer small buffers than we would like (at time of writing, we're waiting for RSTS V7.1), but with enough that we get virtually no "no buffers" messages, and few periods of "no logins".

It appears that the system performance degradation caused by the poorly structured disk resulted in the "loss" of the small buffers. But trying to explain why is difficult.

The number of small buffers used "statically" would have been less after the first disk reorganization, since there were fewer jobs and fewer files open (See "RSTS/E's Small Buffers" by Tim Hart in the RSTS Professional, Vol. 4 No. 1 (Feb '81)). The missing small buffers must have disappeared into dynamic uses. Terminal activity wouldn't have been the culprit as little changed in that area. The problem must be tied to the disks, and especially the rebuilt one. I can only guess that because the disk was poorly structured, the FIP took longer to do its things, and so its queue lengthened. The small buffers disappeared into this queue, and as the small buffers ran out, RSTS slowed down, making matters worse.

If anyone has a more detailed explanation, I would appreciate hearing it.

The moral of this story is that disk organization is a major performance factor, in many subtle ways. Had there been plenty of small buffers, system performance would have degraded with very little indication of why (especially without performance statistics).

One last comment, this time about DSU. The disk it rebuilt had about 300,000 blocks to copy, in 650 or so files. It took in the order of 16 minutes! Very impressive. However, the clustersizes for all files were optimized; this consumed an extra 3500 blocks. DSU is an excellent tool, but it must be used with care; the original disk reorganization was done with DSU also. ♥

LETTERS to the RSTS Pro ...

. . . continued from page 6

Since your last issue came out, our installation has been an avid fan of David Spencer's Standard EDT 2.0 Initializer ("Proposed Standard EDT 2.0 Initializer", pages 74-78, RSTS/E Professional", February, 1982, v.4,#1) [*See also April 1982,v.4,#2, p. 24 "EDT Hints & Kinks"*].

However, we have found some errors and would like to share our solutions to them. In addition, we have a few suggestions to enhance Mr. Spencer's conception, which will at the same time not depart too drastically from the goal of a universal standard.

Bugs:

a) The DELIMITERS - WORD - PROCESSING and DELIMITERS - PROGRAMMING macro names were switched. The latter macro contained the DELIMITERS-WORD-PROCESSING character set, and so the wrong delimiter set was initialized.

Solution: Transpose the lines in each macro beginning with "C; ISE ...".

b) The Screen Width macros did not set the wrap-around width. The result was that the 132-width screen wrapped at position 79.

Solution: Remove the wrap command from the end of the initializer file and put the appropriate one in each macro. (See the attached revised initialization file for details. The "~~/\~~" (wonderful idea!) notes the changes.)

c) The GOLD CONT H and GOLD C keys worked incorrectly if the ADVANCE/BACK switch was set to BACK.

Solution: Explictly define the cursor direction:

GOLD CONT H: "(-C D-C +C UNDC)."

GOLD C: "(+C SEL +W CHGCSR)."

Suggestions:

1. Change GOLD Q to "QUIT." and redefine GOLD CONT A to "QUIT/SAVE." The rationale is that QUIT/SAVE litters the account with usually unnecessary journal files. GOLD CONT A was chosen after GOLD CONT Q and GOLD CONT O didn't work.

2. Disable the HELP command by redefining it to ".". People using the advanced editor generally do not need the HELP option, and it is unfortunately all too easy to hit accidentally. The GOLD HELP key still works, however, so the option is there if needed.

3. Remove the comma from the DELIMITERS-PROGRAMMING set. The comma was an "unexpected" word entity to everyone using the editor here, and was generally deemed less than helpful. We decided that the DELIMITERS-WORD-PROCESSING macro could be tailored to individual tastes, since each account requires its own copy of EDTINI.EDT, while the DELIMITERS-PROGRAMMING macro would be standard.

In general, the standard initializer makes edition a lot more fun, since almost all operations are now accessible through key pad mode. Can't wait to see part 2!

Sincerely, The EDT'ers at DCA, Inc.
Minneapolis, MN

P.S. This letter is an EDT creation.

[*Mr. Spencer's reply follows.*]

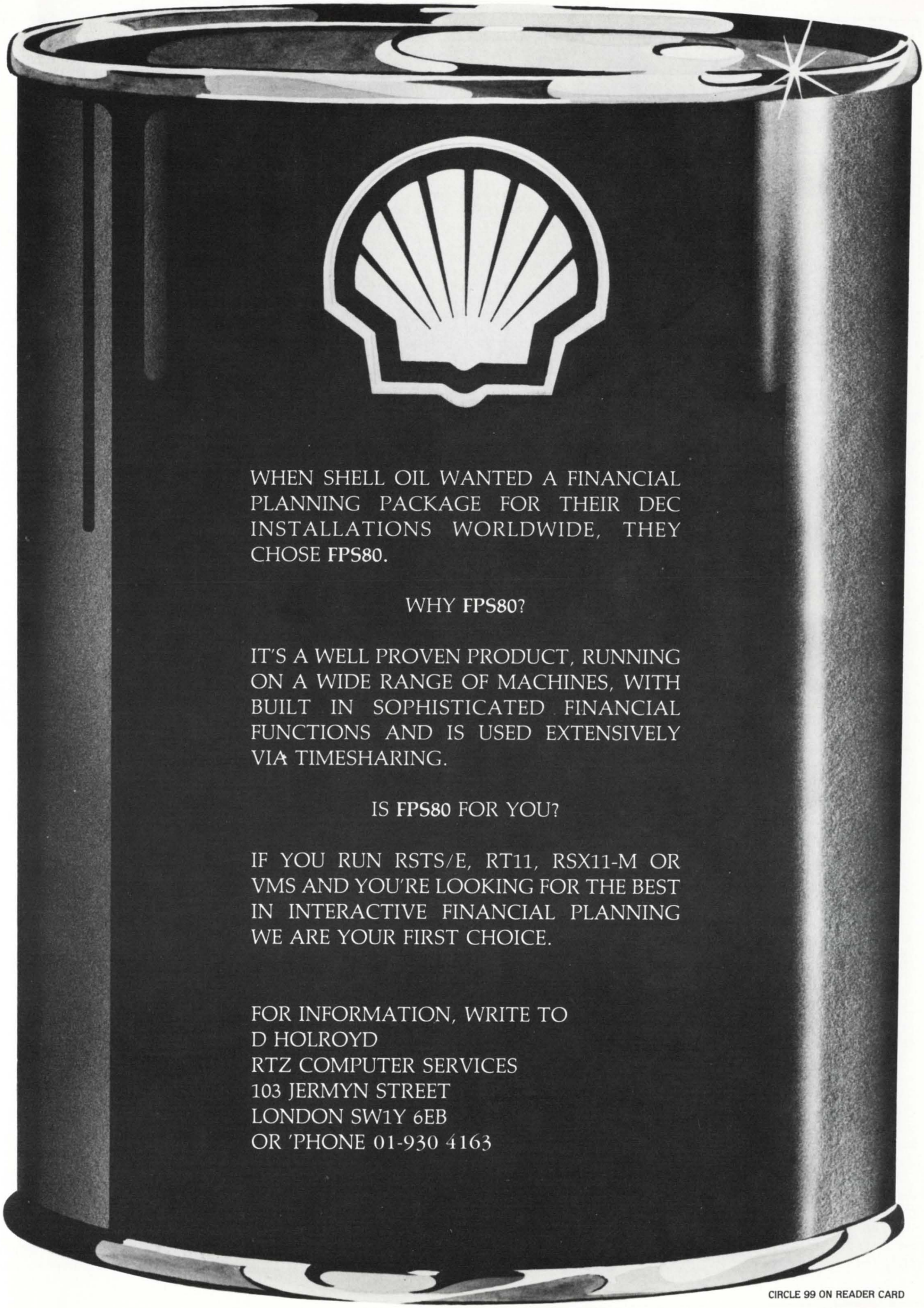
Thank you for the interesting letter. I am of course pleased that somebody took the time and trouble to so thoroughly read and respond to my article.

It is pleasing to see that others are discovering the wonderful world of EDT. The redefinition of the HELP key is a great idea. That and the fix to the screen toggling macros and the GOLD BACKSPACE correction have found their way into my EDT initializer.

As to the word delimiters, it was my intent for the programming set to have all the various math symbols as delimiters. However, your group seems to like only a few delimiters for programming. To each his own delimiter set.

In reference to the other items, my version allows to change case backwards. Your "correction" forces forward changes always. I

```
!                Standard Initializer File        V01.01          10-Mar-82
!+
!                Word Delimiter Macros
!-
DEF M DELIMITERS_WORD_PROCESSING
F=DELIMITERS_WORD_PROCESSING
I
DEF K GOLD CONT D AS 'EXT DELIMITERS_PROGRAMMING.'
^Z
C; ISE EN WO '^Z 9ASC 10ASC 11ASC 12ASC 13ASC 27ASC I ()[],-+*/='^Z EX  ~~/\~~
^Z
!
DEF M DELIMITERS_PROGRAMMING
F=DELIMITERS_PROGRAMMING
I
DEF K GOLD CONT D AS 'EXT DELIMITERS_WORD_PROCESSING.'
^Z
C; ISE EN WO '^Z 9ASC 10ASC 11ASC 12ASC 13ASC 27ASC I '^Z EX    ~~/\~~
^Z
!+
!                Screen Width Macros
!-
DEF M WIDTH_132
I=WIDTH_132
DEF K GOLD CONT W AS 'EXT WIDTH_80.'
SE SC 132
SE WR 131                                                       ~~/\~~
^Z
DEF M WIDTH_80
I=WIDTH_80
DEF K GOLD CONT W AS 'EXT WIDTH_132.'
SE SC 80
SE WR 79                                                        ~~/\~~
^Z
!+
!                Keys Definitions
!-
DEF K CONT B AS '-W.'
DEF K CONT F AS '+W.'
DEF K CONT G AS 'PASTE=?'Put buffer: '.'
DEF K CONT P AS 'PAR.'
DEF K CONT X AS 'CUTSR=?'Cut buffer: '.'
!
DEF K GOLD 12       AS '(-22V).'
DEF K GOLD 13       AS '(+22V).'
DEF K GOLD CONT A AS 'EXT QUIT/SAVE.'                           ~~/\~~
DEF K GOLD CONT D AS 'EXT DELIMITERS_WORD_PROCESSING.'
DEF K GOLD CONT G AS 'CUTSR=DELETE PASTE=?'Rep buffer: '.'
DEF K GOLD CONT H AS '(-C D-C +C UNDC).'                        ~~/\~~
DEF K GOLD CONT W AS 'EXT WIDTH_132.'
DEF K GOLD CONT X AS 'EXT CO SELECT TO=?'Cop buffer: ' ; F L.'
DEF K GOLD CONT Z AS 'EXT EX.'
!
DEF K GOLD . AS 'I~~/\~~^Z -6C.'
DEF K GOLD / AS 'S%~~/\~~%%.'
DEF K GOLD B AS 'EXT F=?'Buffer: '..'
DEF K GOLD C AS '(+C SEL +W CHGCSR).'                           ~~/\~~
DEF K GOLD F AS '(SEL PAR FILLSR).'
DEF K GOLD I AS 'EXT INC ?'Input file: ' =?' Buffer: '.'
DEF K GOLD L AS 'EXT F L.'
DEF K GOLD M AS 'EXT F=MAIN..'
DEF K GOLD O AS 'EXT WR ?'Output file: ' =?' Buffer: '.'
DEF K GOLD Q AS 'EXT QUIT.'                                     ~~/\~~
DEF K GOLD S AS 'EXT SH BU.'
!+
!       Patch to disable HELP option.  dmn 9-Apr-82
!-
DEF K 10 AS '.'                                                 ~~/\~~
!+
!       Set Terminal Characteristics
!-
SE TR
SE K
SE M C
DELIMITERS_PROGRAMMING
!+
!       Set Buffer to MAIN, and Start
!-
F=MAIN
```



WHEN SHELL OIL WANTED A FINANCIAL PLANNING PACKAGE FOR THEIR DEC INSTALLATIONS WORLDWIDE, THEY CHOSE **FPS80.**

WHY **FPS80**?

IT'S A WELL PROVEN PRODUCT, RUNNING ON A WIDE RANGE OF MACHINES, WITH BUILT IN SOPHISTICATED FINANCIAL FUNCTIONS AND IS USED EXTENSIVELY VIA TIMESHARING.

IS **FPS80** FOR YOU?

IF YOU RUN RSTS/E, RT11, RSX11-M OR VMS AND YOU'RE LOOKING FOR THE BEST IN INTERACTIVE FINANCIAL PLANNING WE ARE YOUR FIRST CHOICE.

FOR INFORMATION, WRITE TO
D HOLROYD
RTZ COMPUTER SERVICES
103 JERMYN STREET
LONDON SW1Y 6EB
OR 'PHONE 01-930 4163

have mixed feelings toward your changes to GOLD QUIT. It seems like a great idea for your group, but I prefer training in proper use of the "/RO" switch.

All things considered, I am in favor of personal preferences and think your ideas are great. After all, serving individual needs is what key definitions are all about.

Sincerely, David Spencer

—

Enclosed is my renewal for your excellent magazine. I am impressed with the quality and content of your articles. Keep up the good work.

On another topic, many programs do useless things, but probably the most useless is a program that produces a listing of its own source code. However, such a program is fun to write. I've done one in Basic-Plus which your readers may want to improve upon, expand (or shorten?). Anyway, for what it's worth, here it is. →

```
10      A% = 10%
20      DIM L$(20)
30      L$(1) = "A% = 10%"
40      L$(2) = "DIM L$(20)"
50      L$(3) = "FOR X% = 1% TO 2%"
60      L$(4) = "PRINT NUM1$(A%); TAB(8); L$(X%)"
70      L$(5) = "A% = A%+10%"
80      L$(6) = "NEXT X%"
90      L$(7) = "Z% = 17%"
100     L$(8) = "FOR X% = 1% TO Z%"
110     L$(9) = "PRINT NUM1$(A%); TAB(8); 'L$('; NUM1$(X%); ') ='; "
120     L$(10) = "PRINT CHR$(34%); L$(X%); CHR$(34%)
130     L$(11) = "A% = A%+10%"
140     L$(12) = "NEXT X%"
150     L$(13) = "FOR X% = 3% TO Z%"
160     L$(14) = "PRINT NUM1$(A%); TAB(8); L$(X%)"
170     L$(15) = "A% = A%+10%"
180     L$(16) = "NEXT X%"
190     L$(17) = "END"
200     FOR X% = 1% TO 2%
210     PRINT NUM1$(A%); TAB(8); L$(X%)
220     A% = A%+10%
230     NEXT X%
240     Z% = 17%
250     FOR X% = 1% TO Z%
260     PRINT NUM1$(A%); TAB(8); 'L$('; NUM1$(X%); ') = ';
270     PRINT CHR$(34%); L$(X%); CHR$(34%)
280     A% = A%+10%
290     NEXT X%
300     FOR X% = 3% TO Z%
310     PRINT NUM1$(A%); TAB(8); L$(X%)
320     A% = A% + 10%
330     NEXT X%
340     END

Ready

RUN SHOWME
10      A% = 10%
20      DIM L$(20)
30      L$(1) = "A% = 10%"
40      L$(2) = "DIM L$(20)"
50      L$(3) = "FOR X% = 1% TO 2%"
60      L$(4) = "PRINT NUM1$(A%); TAB(8); L$(X%)"
70      L$(5) = "A% = A%+10%"
80      L$(6) = "NEXT X%"
90      L$(7) = "Z% = 17%"
100     L$(8) = "FOR X% = 1% TO Z%"
110     L$(9) = "PRINT NUM1$(A%); TAB(8); 'L$('; NUM1$(X%); ') ='; "
120     L$(10) = "PRINT CHR$(34%); L$(X%); CHR$(34%)"
130     L$(11) = "A% = A%+10%"
140     L$(12) = "NEXT X%"
150     L$(13) = "FOR X% = 3% TO Z%"
160     L$(14) = "PRINT NUM1$(A%); TAB(8); L$(X%)"
170     L$(15) = "A% = A%+10%"
180     L$(16) = "NEXT X%"
190     L$(17) = "END"
200     FOR X% = 1% TO 2%
210     PRINT NUM1$(A%); TAB(8); L$(X%)
220     A% = A%+10%
230     NEXT X%
240     Z% = 17%
250     FOR X% = 1% TO Z%
260     PRINT NUM1$(A%); TAB(8); 'L$('; NUM1$(X%); ') ='; 
270     PRINT CHR$(34%); L$(X%); CHR$(34%)
280     A% = A%+10%
290     NEXT X%
300     FOR X% = 3% TO Z%
310     PRINT NUM1$(A%); TAB(8); L$(X%)
320     A% = A%+10%
330     NEXT X%
340     END

Ready
```

Bud Dawson, Manager
Technical Support, Computer Operation
MacMillan Bloedel Limited
Vancouver, B.C., Canada

Some time ago, Datamation used to run programming puzzles; this was one of them. We (Univ. of Pennsylvania Medical School Computer Facility) wrote one in FORTRAN and one in TECO! Of course, they were smaller than this — one I think was only one line long!

—

I have been with Digital for a few years and have always been in the hardware line of work. Recently, I have been dealing more with software, mainly RSTS/E. Needless to say, when I found out about the RSTS/E Professional magazine, I was thrilled. It smoothed out the gaps in the documentation and has helped me develop a better understanding of RSTS/E.

Still being hardware oriented, I like to use the software to help find those customer problems that the diagnostics don't always show up, or are so intermittent that the customer can't give up the machine long enough to bring out the problems.

Enclosed are two basic memory exercising programs that I have found useful in bringing out problems while the customer can still use his system for his daily needs. They run detached and at a low priority so not to slow down production more than necessary. I'm sure the customer would gladly help the technician define the problem for an expedient resolution.

The first one is a simple basic program that will only take a few minutes to install. Multiple copies should be run simultaneously to

DILOG GOES WITH LSI-11, PDP-11, VAX-11*

Dilog offers the widest range of single board DEC emulating disc and magnetic tape controllers for LSI-11, 11/2, 11/23, PDP-11 and VAX-11 compatibility.

This growing family includes over 20 software transparent disc and tape products: WINCHESTER AND BACKUP SOLUTIONS FOR MOST APPLICATIONS.

DISC—5¼″ 8″ or 14″ WINCHESTER/SMD/CMD/LARK/CARTRIDGE/FLOPPY controllers with RX02, RK05, RL01/RL02, RP02/RP03, RK06/RK07 and RM02/RM05 emulations and features like 22-bit addressing, 32 or 56-bit ECC, universal formatting (allows you to mix drive types on the same controller without hardware modification), and automatic media flaw compensation.

MAG TAPE—¼″ cartridge/½″ NRZI/PE, streaming or conventional couplers or controllers with TM11, TS03 and TS11 emulations.

NEW VAX-11/PDP-11 CONTROLLERS

- DU 132 TS-11 emulating coupler with expanded buffering for streaming or conventional ½″ industry standard magnetic tape drives. One coupler can accommodate up to 4 drives at speeds to 125 ips, and you get dual density 800/1600BPI as an added feature. RT, RSX, RSTS and VMS software compatibility.
- DU 215 RK06/07 emulating SMD disc controller with 56-bit ECC, universal formatting, optimal device for Winchester and CMD applications. RSX, RSTS and VMS software compatibility.
- DU 218 RM02/05 emulating SMD disc controller for SMD and Winchester applications with full software transparency under RSX and RSTS as well as Media compatibility when used with 80 and 300 MB SMD (CDC 9762/9766) compatible disc drives.

NEW LSI-11 CONTROLLERS

- DQ 212/215 SMD interface. Universal formatting allows mixing or matching two 8″ or 14″ drives with different characteristics and without component changes for up to 220 MB of software transparent formatted capacity. 56-bit ECC, RP02/03 or RK06/07 emulations.
- DQ 444 CDC FINCH interface. Intelligent uP module mixes any two drives of this class with universal formatting. RL01/02 emulations. Built-in drive capacity expansion handling.

All DILOG controllers are price competitive with significant OEM discounts being offered under a Mix and Match plan. 30-day delivery is standard. Distributor inquiries invited. For complete price/performance details, contact DILOG.

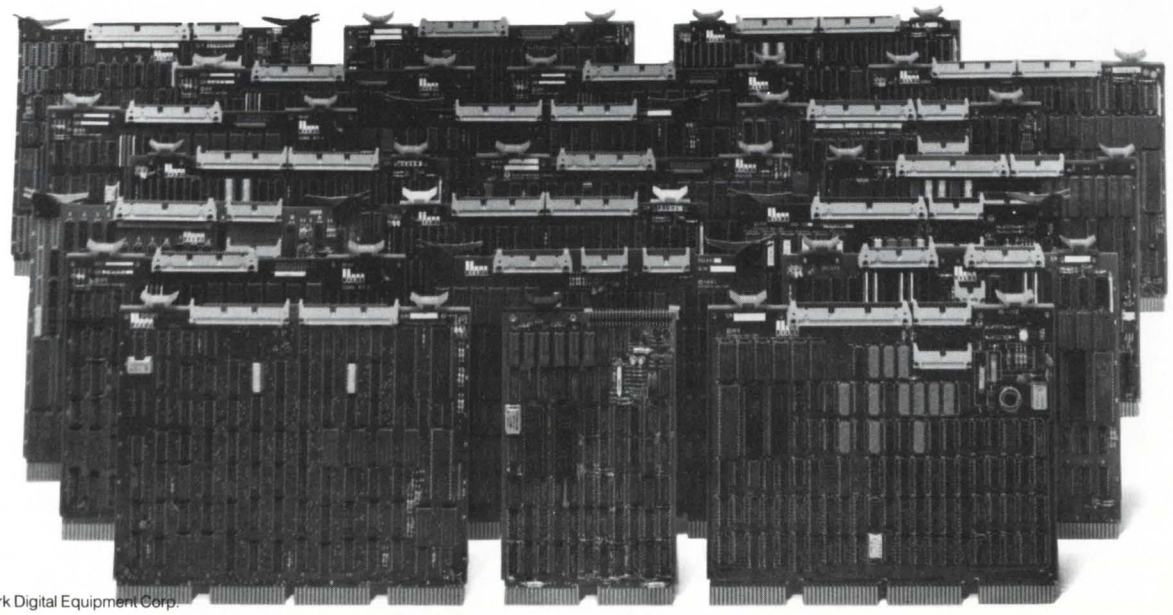
Corporate Headquarters
12800 Garden Grove Blvd. • Garden Grove, Calif. 92643
• Phone: (714) 534-8950 • Telex: 681 399 DILOG GGVE

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

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

DISTRIBUTED LOGIC CORP.
DILOG

NUMBER 1 FOR DEC-11



*Trademark Digital Equipment Corp.

CIRCLE 15 ON READER CARD

bring out flaky memory problems.

The second exercise is more complicated and is in Basic Plus Two. Only one copy of this test should be run. The best thing about this test is it allows you to select the range of memory and the data pattern to test unlike the first exerciser which you have no control of where it resides in memory.

Enclosed you will find all that is needed to implement these exercisers. I hope these programs will be helpful in doing this. It would be nice to hear pros, cons, and suggestions from people who use these programs.

Sincerely, R.A. Smith
NJ District Support
Digital Equipment Corporation

[*Readers: See "Basic Memory Exercising Programs", page 68, this issue.*]

—

We are experiencing some problems with Digital Equipment Corp.—Memphis, relative to the tape drives we are using. We are encountering several problems with our TU16's and TU45's and the local DEC engineering people have asked us to find other sites which share the same problems. We find this hard to believe and think we might be involved in a little DEC run-around scheme.

If you would please find the space necessary to run this in the "Letters ..." of the next issue, it would be greatly appreciated.

For the past 6 years we have been experiencing numerous errors involving our TU16's and TU45 tape drives. Problems such as:

1. Writing a tape and encountering an error 13.
2. Reading a tape and encountering an error 13.
3. Reading a tape on one TU16 with no problem and then reading the same tape on another machine and encountering an error 13.
4. Tape hubs falling off.
5. Tape lock hubs not holding tape reels tight enough.
6. Loss of vacuum.
7. TAPE ERRORS!!!

We have adopted our procedures to accomodate the uncertainty of these tape drives. We now clean all 20 of our tape drives 3 times a day; we now verify every tape we create, (with the equivalent disk storage of 80 RM03's and spinning a tape at only 45 IPS makes this a pretty tedious task); we have to create a copy of each of our permanent storage tapes so that in the event there is a full moon out and a tape which has been used for many months is instantly non-readable we have an image backup of it; and finally, we invested $4,000 in a tape cleaning machine as we were told we had bad tapes (we found this *not* to be the cause).

We have recently asked DEC—Memphis to re-address this problem as we feel we have lived with it long enough. Their response to us was to "find other sites with this problem" as apparently we were the only ones in the country experiencing these problems.

I would ask all interested parties who have learned to live with this nightmare to forward me some documentation about this problem. I, in turn, will pass this newly discovered problem onto our local office with the hopes that DEC—Corporate may soon begin to address the quality problems associated with the TU16 and TU45 (and I would assume the TU10).

Thomas K. Riesenberg
Mgr. Financial Systems & Programming
Baptist Memorial Hospital, Memphis, TN

For some immediate relief, Thomas, see this issue's "DEAR RSTS MAN".

—

Having been an avid consumer since V1:N1, and an occasional implementor of some of your articles, my guilt has caught up with me. In the classic spirit of TIT for TAT, I've enclosed an article which you may find of sufficient interest for your readership to publish. If there is any question, I release the program for copying/usage as your readership sees fit.

You have a superb magazine with broad appeal; I hope it continues in this vein. You are, however, lacking in any serious effort to address the COEM market, especially as it relates to DIBOL. (Yes, there really are CTS500 DIBOL users out there.) Your article by Frank Metcalf (DEC 1981) deserves a reply; there are good, intelligent reasons to choose DIBOL over EVEN Basic Plus 2, which are certainly not evident at first exposure (and blush) to DIBOL. You will find very few professionals who have used DIBOL under RSTS/E for any extended length of time and then abandoned it for another language; the same cannot be said for BASIC/FORTRAN/COBOL(?)...

I would be pleased to submit future articles; having worked with DEC for over 15 years, I would like to share some of this experience.

Robert A. Dudley, President
Meramec Automated Solutions, Inc.
St. Louis, MO

[*Readers: Mr. Dudley's article, "Using the VT100 Printer Port Option Effectively", appears in this issue on page 47.*]

—

The following is a correction to statement four of line 1020 of TIMER.BAS written by Michael H. Koplitz, which appeared in *RSTS PROFESSINAL*", v.3, #4 (Dec. 1981), p.38.

```
1020  TEST.PROJ% = ASCII (MID(Y$,22%,1%))
\       GOTO 1230 IF TEST.PROJ% = 1%
\       TEST.KB% = ASCII(MID(X$,4%,1%))
\       GOTO 1230 IF TEST.KB% AND 128%
                              !PROJECT NUMBER.
                               SKIP IF ACCOUNT
                              [1,*]
                               OR DETACHED.
```

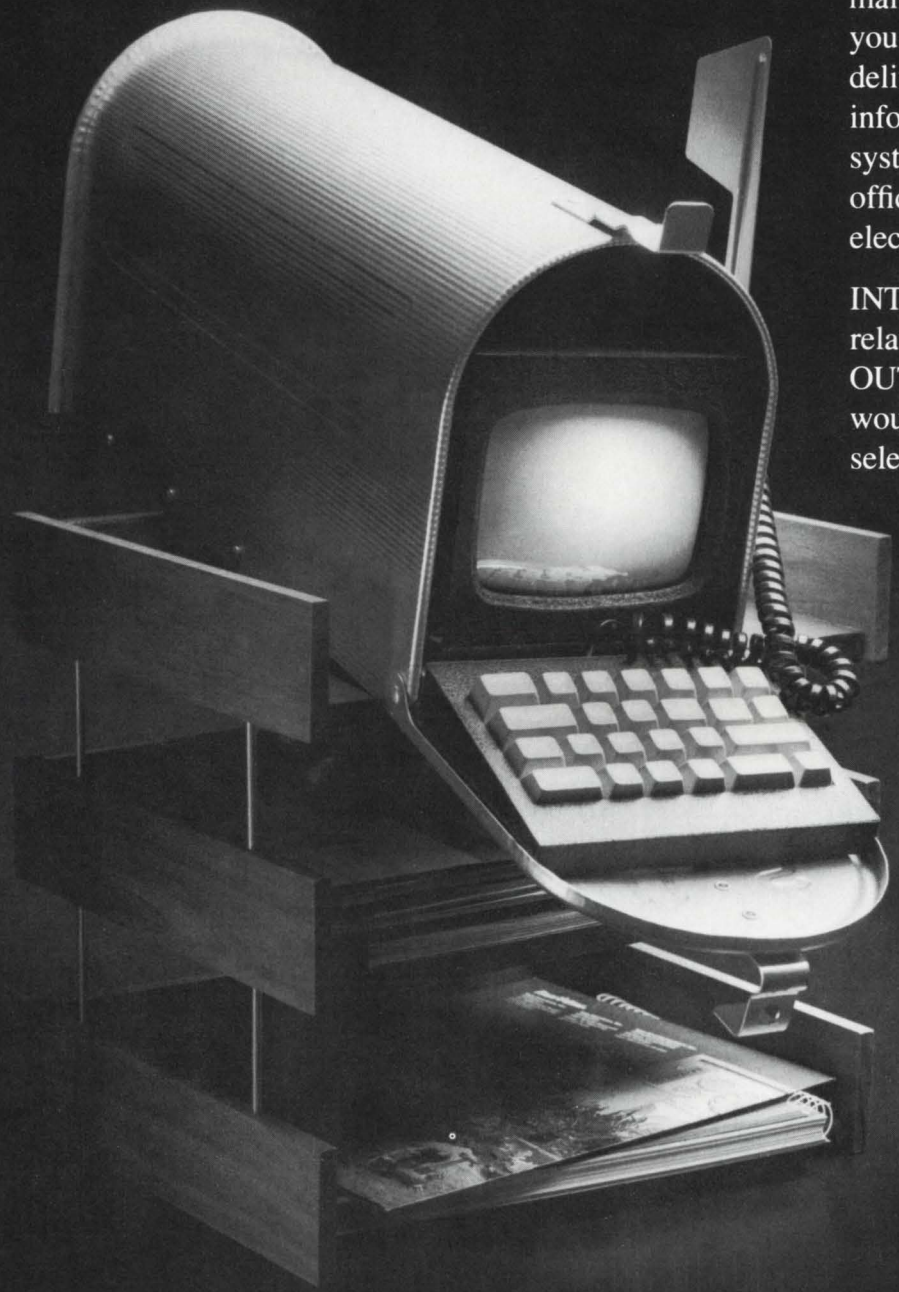
Mr. Koplitz has articles in this issue on pages 37 and 42.

—

EXTRACT! EXTRACT! EXTRACT!

A larger version of Stephen Munyan's "EXTRACT" is available to those interested. Mr. Munyan's article appeared in "RSTS Professional", v.4, #2, April 1982, p. 85. Write to: EXTRACT, c/o RSTS Professional, P.O. Box 361, Fort Washington PA 19034-0361.

ELECTRONIC MAIL. PRACTICALLY SPEAKING.



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

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

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

INTECOM...*the* INtelligent TExt COMmunicator.

NCCS USER 0110

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

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

CIRCLE 76 ON READER CARD

TECO 2

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

All users at Erskine do text editing with TECO. Some users know how to use just a few commands. Others know more. It is easier to learn a little TECO and add to it rather than start with some simpler editor and switch to TECO later.

$TECO2.TEC (listed below) is a file of Erskine macros that is loaded every time a TECO session is started. These macros improve the usability of TECO without adding very much overhead. Each macro will be explained later. It is necessary to modify the file $TECO.TEC by adding @EI/$TECO2/ just before the !DONE! at the bottom if you want TECO to load these macros automatically.

```
                                !Erskine TECO 2 Macros!
@±A/
M* / 96EV V

@±UC% [A ETUA 7ET 155↑T 72↑T 155↑T 74↑T 126↑T 156↑T 12↑T QAET ]A %

@±UD% [A .UA @↑A/Chr?/ ↑T@I// .-1,..XA -D 13↑T 10↑T :@S/↑EQA/''S QA,..D' ]A %

@↑UG% QA,..XA QA,..K %

@↑UH% MC @↑A?For searches:
↑S      Not any alpha-numeric chr          ↑X      Any chr
↑Qc     use "c" literally                  ↑Nc     any character but "c"
↑EA     A-Z, a-z                           ↑ED     0-9
↑EL     line terminator                    ↑ER     Any alpha-numeric chr
↑EQq    Those chr stored in Q-reg q        ↑ES     Any group of spaces/tabs
?%

@↑UQ% @I"~ ~ / \ ~ ~
" @EG'TE/FIND' %

@↑UU% [A [1 .UA @↑A/Chr?/ ↑T@I// .-1,..XA -D
  13↑T 10↑T :@S/↑EQA/''S (.-QA)U1 QAJ Q1< 0A''W 0A+32@I// DR' C >' ]1 ]A %

@↑UV% MC 12V %

@↑UW% [A .UA :@S/↑ES/''S QA,..D ' ]A %

(ET(511-128))ET<ESC><ESC>
```

The first command group prints "M*" to show that the Erskine macros are being loaded, sets 96EV (see EV section of the TECO manual), and displays the first line of the text being edited. The rest of $TECO2.TEC loads various Q-registers with the Erskine commands. The user types M<Q-register name><ESC><ESC> to execute one of the macros. TECO maintains a position pointer between the characters in the buffer. This position pointer will be refered to as "dot".

Erskine TECO 2 macros

MC Clear a scope screen. This macro prints <ESC>H<ESC>J (for VT52), <lead in><chr 28> (for Hazeltine 1400/1500), and <FF> (for ADDS 980/580).

MD Super delete - This macro ask "Chr?" and deletes all text between dot and the first occurrence of the next character typed. Be sure to hit the right key when using this macro!

MG Cut and Paste - This macro requires two steps. Step 1: Move dot to a position that is in front of the first character of the text that is to be cut. Type .UA<ESC><ESC>. Step 2: Move dot just beyond the last character wanted. Type MG<ESC><ESC>. This will move all text delimited by steps 1 and 2 into Q-register A and will delete the text from the buffer. The cut text is pasted into the buffer with the command GA<ESC><ESC>.

MH Help message.

MQ Worry about crash - Marks the file you are editing with "~ ~ / \ ~ ~", exits TECO, and re-enters TECO finding and removing the mark (with TE/FIND). MQ eliminates losing work by having your buffer disappear when the system loses power, etc.

MU Lower case - Like MD but changes all alpha characters from dot to first occurrence of the next character typed to lower case.

MV Snapshot - Will clear a scope screen and display 11 lines above the line containing dot, the line containing dot, as well as 11 lines below the line containing dot.

MW Delete the next word. Deletes from the current position through the next group of spaces/tabs. ♥

BRAEGEN MPD HAS THE DISK STORAGE SUBSYSTEM YOU NEED

INTRODUCING THE DSL-SERIES OF DISK STORAGE SUBSYSTEMS

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

BENEFITS

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

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

Marketing Dept:

BRAEGEN MPD

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

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

*Trademarks of Digital Equipment Corporation

CIRCLE 95 ON READER CARD

BIT AND BYTE MANIPULATION TECHNIQUES IN BASIC+2

By Mark J. Diaz, Dataguard Corp., Hinsdale, IL 60521

Overview

In my experience, many BASIC+2 programmers and analysts are not introduced to the bit and byte manipulation techniques available in BASIC+2.

This article will provide examples and explanations of the methods used to effectivly manipulate bits and bytes within BASIC+2. Also, there is an introduction to both the terminology and the diagrams used in this article.

These techniques are generally applicable to BASIC+ as well.

Why bits and bytes?

Using bit manipulation techniques can result in:

- Smaller data files.
- Shorter data file records.
- Reduced disk accesses.
- Elimination of record sorts or selections.
- Better use of RSTS/E internal flags.
- Shorter program runs.

Unfortunately, these techniques can also result in:

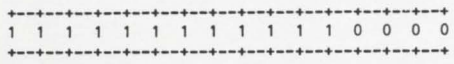
- Some training of maintenance programmers.
- Some loss of transportability (to non-DEC BASIC).
- Some loss of flexibility (Generally only binary (Yes/No) data).
- More difficulty using SORT-11 on bit-encoded fields.
- Possible increased maintenance costs.

What are bits and bytes?

Bits

A bit is the smallest unit of computer storage available. It has two possible states, "on" or "off". Traditionally, "on" is represented by a "1" and "off" is represented by a "0". Twelve of the bits in diagram 1.0 are "on" and four of them are "off".

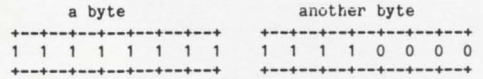
Diagram 1.0



Bytes

A byte is a collection of eight contiguous bits. There are two bytes depicted in diagram 2.0.

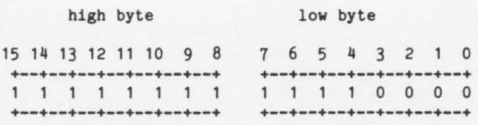
Diagram 2.0



Words

On the PDP-11 a word is a collection of two contiguous bytes, which therefore is sixteen contiguous bits. (On the PDP-11, a word starts at an even location in memory.) As shown in diagram 3.0, the two bytes contained within a word are called the "high byte" and the "low byte". By convention, the bits within a word are numbered from zero to fifteen and from right to left as shown.

Diagram 3.0



Integers are words

In BASIC+2 there is a one-to-one correspondence between a word and an integer. The pattern of bits is a binary (base 2) representation of the decimal (base 10) value of the corresponding integer.

Each bit in a word corresponds to a power of two, starting on the right with bit 0, which equals 2↑0 (1), and ending on the left with bit 15, which equals 2↑15 (32768).

Diagram 4.0 shows the relationship between bits in a word and their associated power of two. The binary representation of a few decimal numbers is shown in diagram 4.1.

Diagram 4.0

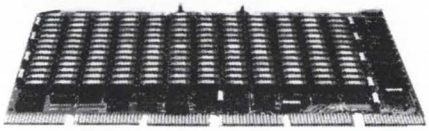
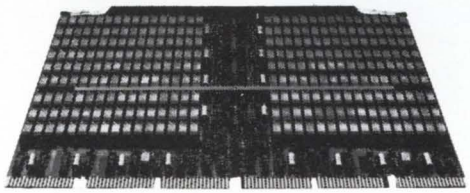
Bit	Power of two
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768

Diagram 4.1

Decimal	Binary
1	0000000000000001
2	0000000000000010
3	0000000000000011
4	0000000000000100
255	0000000011111111

Attention VAX* users:

Before you buy your next DEC* add-in memory compare MTI's price on these direct replacements.

DEC	VAX 11/750	MTI
DEC MS750 AA 256K byte capacity **$7,700**		Intel MU-5750-256 256K byte capacity **$1,445**
DEC MS750 AB 512K byte capacity **$10,200**		Intel MU-5750-512 512K byte capacity **$2,295**
DEC MS780 DA 256K byte capacity **$7,700**	**VAX 11/780**	Intel MU-5780-256 256K byte capacity **$1,445**
DEC MS780 DB 512K byte capacity **$10,200**		Intel MU-5780-512 512K byte capacity **$2,295**
DEC MS780 DC 1024K byte capacity **$13,800**		Intel MU-5780-1024 1024K byte capacity **$3,395**

Save thousands of dollars on VAX add-in memories from MTI. We are authorized stocking distributors for Intel, a leading supplier of add-in memories for DEC systems.

Our Intel boards are direct replacements for DEC memory boards and are completely hardware and software compatible. Our boards perform the same as the DEC equivalents -at a much lower cost- yet are more reliable than any VAX memories on the market today.

MTI has Intel's full line of products. Development systems, chips, single board computers, memory boards and systems. And we are the one source for all the terminals, peripherals, systems, applications expertise and service you'll ever need for your communications network. At prices that are hard to beat. Call MTI today and save.

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

*VAX & DEC are registered trademarks of Digital Equipment Corp.

mti systems corp.

†MTI New York and New Jersey are franchised Intel distributors.

CIRCLE 107 ON READER CARD

Masks

A word that has a known, usually constant pattern of bits is termed a "data mask" or, more simply, a "mask".

Masks find use in testing for the presence or absence of particular bits in other words. Masks also are used to set or clear bits in other words.

Setting and testing bits

Setting bits by direct assignment

Bits in integers can be set by direct assignment. Simply assign the decimal value of the corresponding bit pattern to the integer. This is generally useful for initializing masks. Diagram 6.0 shows the sixteen integer masks BIT.0% to BIT.15% and how they are set by direct assignment.

Diagram 6.0

Assignment			Bit pattern
BIT.0%	=	1%	0000000000000001
BIT.1%	=	2%	0000000000000010
BIT.2%	=	4%	0000000000000100
BIT.3%	=	8%	0000000000001000
BIT.4%	=	16%	0000000000010000
BIT.5%	=	32%	0000000000100000
BIT.6%	=	64%	0000000001000000
BIT.7%	=	128%	0000000010000000
BIT.8%	=	256%	0000000100000000
BIT.9%	=	512%	0000001000000000
BIT.10%	=	1024%	0000010000000000
BIT.11%	=	2048%	0000100000000000
BIT.12%	=	4096%	0001000000000000
BIT.13%	=	8192%	0010000000000000
BIT.14%	=	16384%	0100000000000000
BIT.15%	=	-32768%	1000000000000000

NOTE: Because BASIC+2 uses bit 15 for a sign bit (signed rather than unsigned integers) it is necessary to access it with -32768 rather than +32768

Setting bits by logical operations

Bits are generally set with the logical operator "OR". Diagram 7.0 shows how bit 2 is set in the integer A%.

For your convenience Diagram 8.0 shows the truth tables describing the results of some common logical operations. Section 2.4.1.4 of the BASIC+2 Language Manual shows other available logical operations.

Please note that all logical operations are done on a bit by bit basis.

Diagram 7.0

A%	1111111111110000
BIT.2%	0000000000000100
A% = A% OR BIT.2%	1111111111110100

Diagram 8.0

p	q	p or q
0	0	0
0	1	1
1	0	1
1	1	1

p	q	p and q
0	0	0
0	1	0
1	0	0
1	1	1

p	not p
0	1
1	0

Testing bits

Testing if a particular bit is set in an integer is generally done with the logical operator "AND".

Testing if a particular bit is not set is accomplished by using the logical operators "AND" and "NOT".

Let us analyze which of the following PRINT statements would execute. Assume A% has the value 101101 and the two masks BIT.2% and BIT.4% are defined as would be expected (from diagram 6.0).

We see that when the result of a logical operation is zero it is considered false and the print statement does not execute. When the result of a logical operation is non-zero it is considered true and the print statement is executed. Note that the "NOT" operator takes precedence over the "AND" operator.

```
\ PRINT "Bit 2 is set." IF A% AND BIT.2%

                A%                      101101
                      BIT.2%            000100
                A% AND BIT.2%           000100
                Executes PRINT statement


\ PRINT "Bit 2 is not set." IF NOT A% AND BIT.2%

                    A%                  101101
                NOT A%                  010010
                           BIT.2%       000100
                NOT A% AND BIT.2%       000000
                Does not execute PRINT statement

\ PRINT "Bit 4 is set." IF A% AND BIT.4%

                A%                      101101
                      BIT.4%            010000
                A% AND BIT.4%           000000
                Does not execute PRINT statement

\ PRINT "Bit 4 is not set." IF NOT A% AND BIT.4%

                    A%                  101101
                NOT A%                  010011
                           BIT.4%       010000
                NOT A% AND BIT.4%       010000
                Executes PRINT statement
```

An alternative method exists for testing if particular bits are zero. You may explicitly test the logical AND of the integer and the bit pattern. For example, test as follows to see if bit 4 is zero in the integer A%:

```
\ PRINT "Bit 4 is not set." IF (A% AND BIT.4%) = 0%

                 A%                     101101
                       BIT.4%           010000
                (A% AND BIT.4%)         000000
                (A% AND BIT.4%) = 0%    TRUE
                Executes print statement
```

Monitor Tables

As many locations in the RSTS/E monitor contain bit-encoded status flags, bit testing is essential to the full use of these monitor tables. One such example is the status flags in the Window Control Block (WCB) of a large file system.

Assume the integer WCB% contains the first word of a Window Control Block for an open disk file. We are able to determine many attributes of the file and how it is open by testing bits eight through fifteen of WCB%.

Condition	Test
Opened non-file structured	IF WCB% AND BIT.8%

Read protect against owner	IF WCB% AND BIT.9%
Write protect against owner	IF WCB% AND BIT.10%
Open in update mode	IF WCB% AND BIT.11%
Contiguous file	IF WCB% AND BIT.12%
Current block is locked	IF WCB% AND BIT.13%
File is really UFD	IF WCB% AND BIT.14%
This WCB received original write privileges	IF WCB% AND BIT.15%

For details on RSTS/E monitor tables see the series of articles in previous issues of this magazine, DECUS handouts, and TBL.LST (from the system generation).

BASIC+2 Example

A large amount of inquiry and selection needs to be done to an employee file. As there is an elapsed time constraint, most usual methods proved to costly in terms of disk I/O. But by storing all the relevant data from the employee file in an in-core array, indexed by employee number, elapsed time was reduced to an acceptable level.

The code in diagram 10.0 stores the employee age in the low byte of the array. The employee's sex, marital status, pension eligibility, and whether hourly or salaried are stored in five of the eight bits in the high byte; as defined by diagram 10.1.

It is assumed that the employee's age is not more than 255 (the largest number that can be stored in eight bits). Program size limitations precluded storing separate arrays for each data item.

Diagram 10.0

```
100     \ MAP (EMP)
                EMP.REC$        = 9%

        \ MAP (EMP)
                EMP.NUMBER$     = 3%    ! 001 to 999
                ,EMP.SEX$       = 1%    ! Male or Female (M,F)
                ,EMP.AGE%               ! Employees age
                ,EMP.MARITAL$   = 1%    ! Married or Single (M,S)
                ,EMP.PENSION$   = 1%    ! Yes or No (Y,N)
                ,EMP.TYPE$      = 1%    ! Hourly or Salaried (H,S)

        \ BIT.FEMALE%   = 256%          ! Bit 8
        \ BIT.MARRIED%  = 512%          ! Bit 9
        \ BIT.PENSION%  = 1024%         ! Bit 10
        \ BIT.HOURLY%   = 2048%         ! Bit 11
        \ BIT.VALID%    = 4096%         ! Bit 12
        \ MASK.AGE%     = 255%          ! All bits in low byte.

        \ MAP (EMPARY)                  ! In-core inquiry array
                EMP%(999%)

        \ CALL OPEMP(1%)        ! Open employee file, channel 1
        \ EOF%  = 0%            ! Initialize end of file flag

        \ CALL GETNX( 1%, EMP.REC$, EOF% )      ! Get employee rec
        \ UNTIL EOF%

                \ TMP% = BIT.VALID%     ! Set "valid employee code"
                \ TMP% = TMP% OR (EMP.AGE% AND MASK.AGE%)
                \ TMP% = TMP% OR BIT.FEMALE%    IF EMP.SEX$     = "F"
                \ TMP% = TMP% OR BIT.MARRIED%   IF EMP.MARITAL$ = "M"
                \ TMP% = TMP% OR BIT.PENSION%   IF EMP.PENSION$ = "Y"
                \ TMP% = TMP% OR BIT.HOURLY%    IF EMP.TYPE$    = "H"

                \ EMP%(VAL%(EMP.NUMBER$)) = TMP%

                \ CALL GETNX( 1%, EMP.REC$, EOF% )

        \ NEXT
        \ CLOSE #1%
```

Diagram 10.1

```
15 14 13 12 11 10  9  8    7  6  5  4  3  2  1  0
 +--+--+--+--+--+--+--+--+  +--+--+--+--+--+--+--+--+
 X  X  X  V  H  P  M  S     ----Employee's age----
 +--+--+--+--+--+--+--+--+  +--+--+--+--+--+--+--+--+

      X = Not Used
      V = Valid Employee
      H = Hourly
      P = Pension
      M = Marital Status
      S = Sex
```

As a program progresses it need only make array references to test if any employee is married, single, male, female, etc. rather than making random file accesses.

For example; given that EMP.NUM% is an employee number:

```
\ EMP.DATA$ = EMP$(EMP.NUM$)          ! Get employee data

\ PRINT "Employee has pension"    IF     EMP.DATA$ AND BIT.PENSION$
\ PRINT "Employee is single"      IF NOT EMP.DATA$ AND BIT.MARRIED$
\ PRINT "Not an employee number" IF NOT EMP.DATA$ AND BIT.VALID$
\ PRINT "Employee's age is" ;           EMP.DATA$ AND MASK.AGE$
```

Clearing bits

You may need to clear a particular bit in an integer. The following example shows a method for clearing bit 7 in the integer A%.

```
\ A$ = A$ AND NOT BIT.7$          ! Clear bit 7 of A$

                  BIT.7$          0000000010000000
              NOT BIT.7$          1111111101111111
          A$                      0110001111001010
          A$ AND NOT BIT.7$       0110001101001010
```

Byte oriented operations and bit patterns

We have already seen one example of a byte oriented operation. In the employee file example (Diagram 10.0) we used the low byte to store an employee's age. Then the age was retrived by ANDing the proper array element with a bit pattern whose low byte was all ones (255).

The high byte can be accessed in a similar manner by first swapping the high and low bytes and then ANDing with 255. For example:

```
          A$                    00111110 10000001
SWAP$(A$)                       10000001 00111110
              255$              00000000 11111111
SWAP$(A$) AND 255$              00000000 00111110
```

A common example of using each byte of an integer separately is the RSTS/E internal format for an account number (PPN% in this example)

```
\ PPN$          = PEEK(PEEK(PEEK(520$)+8$)+24$)
\ PROJECT$      = SWAP$(PPN$) AND 255$
\ PROGRAMMER$   = PPN$ AND 255$
```

Clearing bytes is easily peformed as follows:

```
\ A$ = A$ AND 255$              ! Clear high byte
\ A$ = A$ AND NOT 255$          ! Clear low byte
\ A$ = 0$                       ! Clear both bytes
```

Hints and considerations

I would suggest using some convention to identify an integer as a bit pattern of a data mask, such as prefixing all masks with "BIT." or "MASK." I use "BIT." for single bit masks and "MASK." for multiple bit masks. Although the fact that you are doing a bitwise logical operation is determinable from context, using a standard prefix will make this readily apparent.

As programming in this manner can result in somewhat less maintainable code, I refrain from using these techniques unless they are required for the success of the situation at hand. There is certainly no advantage in reducing four integer flags in a program to one bit-encoded integer flag. That is, it is certainly less efficient and less maintainable to code "IF STATUS.FLAG% AND BIT.EOF%" rather than the more straight forward "IF EOF%"

It is often desirable to use two bits for some binary valued data. Using a previous example, if BIT.MARRIED% and BIT.SINGLE% were defined to be different bits, and the coresponding bits in the array EMP%() were set, you could test for married and single more conveniently. As follows:

```
\ PRINT "Married" IF EMP%(EMP.NUM%) AND BIT.MARRIED%
\ PRINT "Single"  IF EMP%(EMP.NUM%) AND BIT.SINGLE%
```

Review of advantages

Smaller data files—shorter data file records

One integer could be used to replace sixteen one byte binary flags, an eight to one reduction. This might be appropriate if space were critical enough to warrent the extra programming effort. Shorter data file records are generally processed more efficiently than long data records.

Reduced disk accesses — Elimination of record sorts or selections

Keeping an in-core array instead of randomly accessing a data file for each desired data record will result in less disk I/O if sequentially reading the file once results in less activity than all the random accesses.

One example where reduced disks accesses would be realized is the following case, a one-shot conversion from another system.

1) A large input file (multi-volume tape) is to be read sequentially.

2) One (non-key) field of the input record is the employee number.

3) The records are to be processed differently based on the contents of the associated employee record (not the input record).

To avoid many passes over the input file and constant random accesses of the employee file, an in-core bit-encoded array was built of the pertinent employee information. This eliminated the random accesses on the employee file for each input record and involves only one pass on the input file.

Review of disadvantages

Remember that if these methods are used, you may experience any or all of the problems outlined in the introduction; but that when warranted and used with prudence, these techniques can greatly increase the capabilities of your system.

CLASSIFIED

Send Classified Ads to: RSTS Classified, P.O. Box 361, Ft. Washington, PA 19034-0361. 1^{00} per word, first 12 words free with one year's subscription. [Be sure to include a phone number or address in your message.]

RSTS Heavyweights Available. Guaranteed results. SYSGENS, Troubleshooting, Optimization, Disaster Recovery, Consulting (203) 683-2360

DIGITEC Software Design, Inc. A Coordination of Computer Software Sciences. (206) 821-7507

CUSTOM SOFTWARE — Tridaco Systems Inc. manufacturing, point-of-sale, purchasing.

RSTS CHEAP!

$350/PORT, $20/MB

on **WALL STREET, NYC**

Local error-free dial-in
or leased line termination

and in **PHILADELPHIA**

DEC 11/70, 24 hrs, 7 days
Secure Backups
B+2, DIBOL, COBOL81
DATATRIEVE, WORD-11, EDT
Secured by **LOCK-11**

NATIONWIDE DATA DIALOG
DAVE MALLERY **215-364-2800**

RSTS RESCUE SQUAD

We salvage all kinds of disasters:

- unreadable disks
- ruined UFDs and MFDs repaired
- immediate response
- telephone DIAL-UP
- on-site
- software tools
- custom recovery
- 90% success to date
- more than 1 GB rescued to date

Brought to you by
On Track Systems, Inc.
and a well known (and read)
RSTS expert.
CALL 24 HOURS
215–542-7133

LOOKING FOR DEVELOPMENT TIME?

$7 PER HOUR CONNECT TIME

NO KILOCORE TICK CHARGES
NO CPU CHARGES
RSTS/E TIME

BASIC PLUS
BASIC PLUS, PASCAL, "C" } WITH CROSS COMPILER SUPPORT

WORD-II WORD PROCESSING
WAFE, TECO, EDT } PROGRAM EDITING

BUDGET BYTES 212-944-9230

OMNICOMPUTER, INC.
1430 Broadway, New York, N.Y. 10018

DEC BEST VALUES

PRE-OWNED DEC EQUIPMENT
BUYING AND SELLING
SYSTEMS • CPU's • PERIPHERALS • TERMINALS
OPTIONS • MEMORY • COMPATIBLES

CALL DICK BAKER (305) 979-2844

dataware incorporated
Carico Center
2845 NW 62nd Street
Ft. Lauderdale, Florida 33309
Telephone (305) 979-2844

GROSSMAN GRAPHICS
5041 frankford phila., pa. 19124
215 537-0782

The *FAMOUS* RSTS PROFESSIONAL TEE-SHIRT is now for sale!

Send size desired and $6.95 for each shirt to:
RSTS TEE-SHIRT
P.O. Box 361
Ft. Washington, PA 19034-0361
Shirts available in adults sizes only:
Small - Medium - Large - X-Large

Certified Data Processing Inc. Computer Services 871-0202 Rockland, Mass.

300 & 1200 baud dial-up timesharing; 11/70 with Basic-Plus and Basic-plus-2. (303) 893-3123. Zimmerman Metals, 201 E. 58th Ave., Denver, Colorado 80216

CTS300 Software A/R, A/P, O/E, G/L, I/C, P/O, PHONE (415) 930-9932 For Info.

Had a crash? Precision Methods Inc. Repairs most types of removable disk packs. (603) 898-2600

For sale. 4 RKO5JAA'S. With controller RK11-d, 66 packs, make offer Jim Canfield. (612) 944-2121

Anyone know how undocumented Dibol XCALL S4S Works? Call Andre Agoos (617) 361-4022 Collect.

Need expert, specialized, personalized, professional, proficient, masterful, meaningful, instruction? See Elaine Edwards.

Prevent RSTS penetrations, hackers, unauthorized access. Read our report! Call (213) 462-5788

Construction; Property Management; Basic-Plus 2/RMS-11K PI consulting group. (213) 654-2693.

Vax System manager seeking employment. Contact Ted at (404) 874-0005.

DEC PDP-11 Series Word Processing with mailing list. (512) 226-8392, CARDA information.

PDP 11/34, Heath H11A, and etc. for sale. Contact David Smith (713) 688-9835.

Dibol-Based Medical Clinic System for sale. ABACUS Computer Corp (304) 465-0921.

Dimension Data Systems provides DEC configured systems at a savings. Call (214) 262-8201

DEC Line Printer Ribbons at significant savings. Call Lori at (203) 327-9210.

Menu-11 is coming!

Skypek, Tener & Associates Mini computer services. Consulting/systems/programs Atlanta area (404) 493-3939.

Russ Ellett & Associates. PDP-11 hardware/software consultants. Systems design and development. (213) 907-6541.

RJM05 Disk Subsystem Equiv. (CDC/EMULEX) $18,900, Butler Assoc. Inc. (617)965-3210.

Still $200! MORE FEATURES!

$200. RSTS/E* WORD PROCESSOR CBEDIT.BAS

Basic-Plus* program with VDT input, window edit and document save. Add, locate, change, replace, delete, block move and copy merge, etc. Crash and operator error recovery. Supports DEC, Hazletine and Mime standard VDT's. Others easy to add.

Fully formatted output (margins, justify, center, underscore, super-sub-script, readers, page numbers, etc.), to typewriter, line printer or disk. Bidirectional driver for Diablo-Xerox 16-17.

User's manual and source code included.
9-Track $200. RKO5 $260. ppd
T. F. Hudgins & Associates, Inc.
P.O. Box 10946, Houston, Texas 77018
Woods Martin 713/682-3651
*TM Digital Equipment Corporation

PLAN PLUS FINANCIAL PLANNING SYSTEM

RSTS/E, $4,500.
PLAN-PLUS LIMITED
P.O. Box 327
London, W4 4QD
Tel. UK 01 834 7334, Ext. 7935
[24 Hours]
Telex 8813271 GECOMSG

MJ-11-AA Memory System for sale. Best offer. Pikes Peak Library. (303) 473-2080.

Microprocessor Cro Software, the most economical approach to programming micros. All popular chips supported. Inquire: Microtec, Ph (408) 733-2919, PO BOX 60337, Sunnyvale, CA 94088.

We buy and sell new and used DEC computers and peripherals.Call (713) 445-0082.

2-RK05 Drives with controller DEC maintained $6500 packs $150 would separate. Richard Ludden (918) 333-6151.

DEC PDP-11/34, 256K, 4 drives, and other gear.(213) 748-5415, Mr. Lister.

Used computer equipment. Washington & Jefferson College. (412) 222-4400 ext.336. Mary Deml.

DATA TRANSMISSION SPECALIST, job wanted. K. Ziemba, Kolberga 8/1 Wroclaw, Poland.

CONSULTANT- 9 yrs. Dibol. Available after Sept. NY area (201) 731-2208.

Looking for Job Shop Control/Job Costing Software. Contact Iseli Company (203) 757-9281.

Need info on San Francisco area RSTS users. Hiring? Write C. Allen P.O. Box 2315, Detroit, MI 48231.

ERGO; Elite Reconstruction, Gaming, and Optimization for RSTS and RT11. Wade (714) 968-2133.

Infinity Software proudly announces INVADE and PACKER! See page 15.

Wanted Analyst/Programmer with RSTS experience to be part owner of 3 year old Knoxville, TN micro-computer software co. No investment required. Computer Assisted Services (615) 691-1515.

SAVE MEMORY with TYPE — 1K RTS, 1K user. Efficiently outputs disk files to terminals. 800/1600 BPI magtape. $100.00. Erskine Software, Box 86, Due West, SC 29639 (803) 379-8816.

Have DZ11E. Need BASIC-Plus programmer with 5 years experience. Small college EDP. (507) 663-4282.

NEWS RELEASES

Occasionally we are requested to print news that may be of interest to the RSTS community. We are happy to offer this feature to our readers. We reserve the right to print only as time and space permit. We cannot return photos or manuscripts. Send news releases to: RSTS News Release, P.O. Box 361, Ft. Washington, PA 19034-0361.

June, 1982

NEW INTELLIGENT ¼" 3M CARTRIDGE MAGNETIC TAPE CONTROLLER INTERFACES UP TO TWO KENNEDY 6450 DRIVES TO LSI-11 CPUs AND PROVIDES UP TO 17MB EACH ARCHIVAL STORAGE FOR DEC OPERATING SYSTEMS

Garden Grove, CA — A new lost cost uP based single board ¼" 3M Cartridge magnetic tape Controller, providing interface for one or two Kennedy 6450 ¼" DE300 type cartridge tape drives with up to 17 megabytes (each) data storage, is now available from DILOG (Distributed Logic Corp.), Garden Grove, California for back-up use with DEC LSI-11, 11/2 and 11/23 computers.

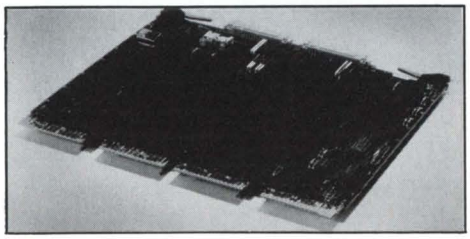
Designated the Model DQ330, the fully self-contained quad size controller requires only a single CPU card slot and ribbon cable for interconnection. it is compatible with DEC TM-11 and TS-03 software drivers in RT-11 and RSX-11 operating systems.

In operation the controller handles read after write Serpentine head drives. The DQ330 includes a diagnostic routine and automatic self-test which causes on-board diagnostics to run each time the Q-bus is initialized. There's also an integral LED that provides indicatio to insure protection of critical data base transfer.

The controller also includes several features for operator convenience * FIFO buffer for DMA latency and memory addressing to 128K words.

Performance data includes 30 ips read/write speed * 6,400 bits per inch format densities and 192,000 bits/sec data transfer rate.

Price: $1,436.00 qty. 50; Delivery: Stock to 45 days A.R.O. Direct inquiries to: Mr. Dennis Edwards, Nat'l. Sls. Mgr., DILOG, 12800 Garden Grove Blvd., Garden Grove, CA 92643. Phone: (714) 534-8950.



DILOG (Distributed Logic Corp.) introduces a new single board ¼" 3M cartridge magnetic tape controller that interfaces one or two Kennedy 6450 drives to a single slot of LSI-11 CPUs, providing up to 17 MB back-up per drive with DEC RT-11/RSX-11 operating system compatibility and TM-11/TS-03 software drivers.

May, 1982

RJ-11 COBOL COMPILER FROM EEC

Wayland, MA — EEC Systems announces a new release of its RJ-11 Cobol Compiler. The new release includes several improvements, but the RUNTIME system is no larger than the previous release. Programs compiled with the previous compiler will run with the new Run-Time system provided they do not use expressions or the INSPECT, STRING or UNSTRING verbs.

The most important feature of the new release is its timesharing facilities. RJ-11 timesharing COBOL supports up to six terminals running COBOL programs concurrently. Programs remain resident while they are operating, so that terminal response times are exceptionally fast, even with floppy-disk systems.

Two or more terminals operating the same program require only one copy of PROCEDURE DIVISION to be loaded. Also, compiled programs are very concise, with most COBOL verbs requiring only 2,3 or 4 PDP-11 words. Hence the system can accomodate for example one large program of several thousand lines, or up to six medium sized programs of several hundred lines each.

Programs to be run under the timesharing system must be in compiled object (SAV) form. The same compiler is used as for single user RT-11 systems so that compiled programs may be run either stand-alone or timesharing without recompiling.

This release of RJ-11 Cobol utilizes a system "File-Specified" file, FSPEC.MAS, having a COMP-3 key rather than the previous straight numeric key. Hence, FSPEC.MAS item numbers can now range up to 999999 rather than 999.

Among other enhancements are: 'WRITE ... ADVANCING' is now allowed to disk files. This allows print-files to be written to disk and later printed. Also certain fast operations on binary disk have been implemented. If for example the data-name I was defined as a binary (COMP) item e.g. 01 I PIC 99 COMP — then each of the following operations will now execute approximately 4 times faster.

1. Accessing an element in an array, if I is the subscript
2. Adding a single digit numeric literal to I
3. Comparing I with a numeric literal
4. PERFORM ... VARYING where I is the 'VARYING' item.

Along with the enhancements is a completely new manual. A spokesman for EEC Systems said that the documentation for RJ-11 Cobol was excellent.

RJ-11 Cobol is an ANSI '74 enhanced compiler available on the RT-11 TSX-Plus, and RSTS/E operating systems for DEC LSI and PDP-11 computers. It requires at least 16 K words of memory. It has a number of special features including multi-processor interactive operation, flexible facilities for processing files, listings showing DATA and PROCEDURE addresses, powerful error locating facilities. RJ-11 Cobol also executes several times faster than DEC COBOL.

Prices for RJ-11 Cobol start at $2,500 with generous OEM and multiple CPU discounts.

More information can be obtained from: EEC Systems, 286 Boston Post Road, Wayland, MA 01778; (617) 358-7782.

May, 1982

FINALLY—A TOTAL MAGTAPE INDEXING AND FILE LOOK-UP SYSTEM FOR DEC COMPUTERS

Kirkland, WA — DIGITEC Software Design, Inc. is proud to announce TAPE-DEX, a long awaited tape-indexing and file loop-up package for DEC computers. The TAPE-DEX system dramatically simplifies magtape indexing and file look-up. An extensive ON-LINE HELP feature makes TAPE-DEX easy to use. TAPE-DEX allows indexing of up to 32,767 magtapes can be used to catalog existing tapes. TAPE-DEX is compatible with DEC's standard backup package, Data Processing Design, Inc.'s SAVER, and DOS-formatted tapes.

The TAPE-DEX package consists of two components, TAPE-DEX I and TAPE-DEX II which may be purchased individually or as a package. Both support indirect command files at all levels.

TAPE-DEX I provides the user with a listing of tapes sorted by any combination of the following fields: date entered, expiration date, label, volume, type, owner, description, and name.

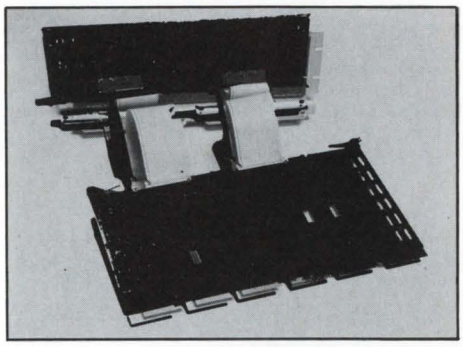
TAPE-DEX II allows the user to locate tapes containing a specified file or files. The user may locate tapes containing all occurances of a particular file, the most recent back-up of a file, the first occurance of the file, or the most recent modification of the file. The user may indicate up to 10 different file specifications (including wildcards). TAPE-DEX II adds catalog information to SAVER and Backup listings.

The TAPE-DEX system is a cost-effective, timesaving system management resource-perfect for both in-house and timesharing environments. The TAPE-DEX package is currently available at an introductory price of $425. For more information about TAPE-DEX and other SYSTEMS MANAGEMENT RESOURCES, please contact: DIGITEC Software Design, Inc., 14125 - 108th Avenue N.E., Kirkland, WA 98033; (206) 821-7507.

June, 1982

EMULEX INTRODUCES DEC DZ11 COMPATIBLE MULTIPLEXER; REDUCES PRICES ON ITS ENTIRE CS11 AND SC21 LINES

Santa Ana, CA — In a pair of related announcements, Emulex Corporation has introduced the third member of its CS21 family of communications multiplexers and also has reduced prices on both the CS11 and CS21 multiplexer families.



The Emulex CS21/Z Series, a single board and passive 16-line distribution panel provide DZ11E replacement to any PDP-11 or VAX-11 computer.

The new model CS21/Z emulates the DEC DZ11E 16-channel asynchronous multiplexer. It employs standard CS21 series hardware with a new firmware set to provide a software transparent emulation of the DZ11E programmed I/O communications function.

The CS21/Z provides the capability to connect up to 16 asynchronous RS-232 communications lines to any PDP-11 or VAX-11 computer, using just one slot in the Unibus backplane. Its standard features include double depth (128 characters per 8 lines) receive FIFO buffering and modem control. The CS21/Z is list priced at $3,500 for a 16-channel system.

Concurrent with introduction of the new multiplexer, Emulex has also reduced prices on the DH11 compatible versions of its CS11 and CS21 series communications products. The DH11 compatible CS21/H has been reduced by 15% in list price for $4,100 to $3,500 which is the same price as the DZ11 compatible version.

"PDP-11 users are no longer faced with the difficult decision of selecting between price and performance because in our CS21 line, the DZ11 and DH11 versions are both offered at the same price," said Phillip (Flip) Begich, director of national marketing. "Given the fact that both types are supported by DEC software on the PDP-112 series, those users should generally pick the higher performance of our CS21/H which emulates the DH11."

VAX users also have the option of gaining significant performance improvements with our DH11-compatible models at a very modest $450 list price increase," he continued. "The DZ11-compatible CS21/Z is still available at the $3,500 list price for VAX users who are highly cost conscious and/or who prefer to use standard VMS software."

Pricing of the CS11 series multiplexers has also been revised for the DH11 compatible models, and certain features which were previously extra-cost options are now included in the basic price. The CS11/H for PDP-11 CPU's is now listed at $4,500 for a 16-channel system, including double-depth FIFO and full 64-line expansion capability, representing a 22.5 per cent reduction from the previous list price of $5,800.

The list price of the CS11/U for VAX-11 CPU's has been reduced more than 20 percent to $4,950, including the Emulex VMS/UH software package, from the old price of $6,250 for a 16-channel configuration.

"All multiplexer versions carry mix/match pricing and can be combined with other Emulex products, such as disk and tape controllers," Begich added. "This allows many of our customers to buy all our products at high volume prices. For example, the CS21/Z or CS21/H price is only $2,520 for 16-channels at the 100 unit price level. Attractive quantity discounts to end users, as well as discounts for Educom members and other educational/governmental organizations, also are available."

The new CS21/Z model plus restructured pricing rounds out our product line to include DZ11, DH11, and DV11 compatible units, and users now have a sound basis on which to select particular product lines and models for their application", Begich said. "The CS21 family is generally cheaper and better suited for smaller installations involving 16-32 channels, whereas the CS11 series is optimum for larger numbers of lines. The two families have a price crossover at 48 channels, with the CS11 becoming less expensive above that point."

The CS11 series, introduced in early 1981, is designed to handle large system configurations. The single CC11 controller handles 8-64 channels and emulates up to four separate functional DH11 units. This reduces backplane space and keeps internal CPU power drain to an absolute minimum. All line adapter circuitry is contained in the channel adapters on the distribution panels. Adapter types, such as RS232 or current loop, can be mixed on the same basis, giving the user considerable flexibility in configuring or adding to a system. Also, troubleshooting can be accomplished on-line for any 8-channel group without affecting the rest of the system. Another advantage to many users is the ability to configure up to 64 lines on a pair of CC11 controllers to in effect have a "hot" backup capability. In the event of a controller failure, all 64 channels can immediately be shifted to the other controller simply by moving cables. Finally, the CS11 series includes full DM11 compatible modem control which permits operation in either full or half duplex modes with split input/output speed flexibility.

The CS21 series is generally optimum for smaller system configurations since all 16 channels of line adapter circuitry are contained on the CC21 controller board. The distribution panel is a simple 5¼-inch high passive unit which is plug compatible with the standard DEC H317 unit. This permits users who already have DZ11's installed to replace the DEC controller without having to rewire the terminal distribution panel. Modem control compatible with the DZ11E is included, and the unit offers switch-selectable single or extended receive FIFO capacity.

Emulex Corporation, headquartered in Santa Ana, is the leading independent manufacturer of disk, tape and communications controllers for interfacing peripheral equipment to computer systems made by Digital Equipment Corporation.

May, 1982
NEW SOFTWARE MAKES FOUR TERMINALS OUT OF ONE

Bedford, MA — Clyde Digital Systems, a division of Clyde Enterprises, Inc., announces a stand alone software package for the DEC PDP-11 and VAX-11 computers that extends a single computer terminal to four interactive terminals. It works for any kind of terminal. The session context can be swapped from one job to another without interruption. This may be done at any point in an interactive session. In addition, all key-strokes entered by the user and all information that is presented to the terminal by the computer are captured in a log file. The key-strokes entered by the user are underlined in the log file. Other powerful control modes are also provided.

Product information may be obtained from: Janet (617) 275-6642.

November, 1981
DIRECT ANNOUNCES VP800/C VIDEO DISPLAY TERMINAL

Sunnyvale, CA — DIRECT, Incorporated of 1279 Lawrence Station Road, Sunnyvale, California recently announced the introduction of its VP800/C video display terminal. The unit is the culmination of the company's [illegible] of ANSI compliant products which includes the previous VP800/A and VP800/B models.

The VP800/C is a programmable video display station which is code compatible with Digital Equipment Corporation's VT100 and VT52 models. The terminal's control codes comply with ANSI Standard X3.65-1979. Stephen Auditore, Director of Marketing for DIRECT, stated that, "The VP800/C provides capabilities attractive to the data entry, buffered editing, and program development markets."

The terminal comes with 16K bytes of display RAM which can be upgraded to 32K. Also included are a buffered printer port, buffered editing, block mode, forms mode, extensive field attributes, and the ability to download programs or sections of programs from the host to the terminal's memory. Downloaded programs can be debugged and run at the terminal, thus easing the load on the host processor.

The terminal's editing feature include erase line, erase page, insert character, delete character, insert line and delete line. For ease of data entry into forms, the VP800/C provides protected, transmit-only, numeric-only, and alpha-only fields, as well as unprotected fields. The terminal will automatically fill fields with zeros or spaces if desired.

According to Auditore, "With the VP800/C's field definition and data checking capabilities, as well as its download program and debugging capabilities, DIRECT now offers a powerful tool for the sophisticated user."

For more information, contact the DIRECT marketing department at (408) 734-5504.

May 1, 1982
SOFTWARE TECHNIQUES, INC. ANNOUNCES A-PLUS ACCOUNTS PAYABLE MODULE

Los Alamitos, CA — Software Techniques Inc. today announced the release of A-PLUS, a modular, integrated financial applications system designed to solve the business accounting problems of companies in a wide variety of industries.

A-PLUS runs on PDP-11 computers under the RSTS/E or CTS500 operating system. Because of its highly efficient design, A-PLUS is ideal for use on small systems or systems with existing application loads.

Accounts Payable, the first A-PLUS module available, manages payables, tracks costs, maintains vendor history, and assists in financial planning. The design objectives for this system were:

- Comprehensive Accounting Capability
- Ease of Use
- Low Maintenance Costs
- Long Usable Life
- Complete Documentation

Rick Scherle, president for the company, says, "What we've done for disk optimization, we're doing for business accounting software. The A-PLUS Accounts Payable module is just part of a total accounting concept that Software Techniques plans to make available to DEC users."

Software Techniques, Inc., headquartered in Los Alamitos, California, is one of the world's leading minicomputer consulting groups. Specializing in Digital's RSTS/E and VMS operating systems, Software Techniques provides products and services world-wide, ranging from business accounting software packages to high-technology consulting services.

April, 1982
DIRECT ADDS THREE DISTRIBUTORS

Sunnyvale, CA — DIRECT Inc., manufacturer of Intelligent Video Display Terminals and work station products, has announced the addition of three distributors to their North American Distributor Network.

ANNESE ASSOCIATES, Inc. with offices in Herkimar, Rochester, Syracuse and Albany, New York will cover the upstate New York area. DYTEC DISTRIBUTORS, Inc. will cover Nebraska, Iowa, Kansas and Missouri from their office in Maryland Heights, Missouri and Lenexa, Kansas. PACIFIC NORTHWEST ELECTRONICS (P.N.E.) with offices in Bellevue, Washington and Portland, oregon will cover Oregon, Washington and the Idaho panhandle.

For more information, contact DIRECT, 1279 Lawrence Station Road, Sunnyvale, CA 94086; (408) 734-5504.

March, 1982

PARALLEL PASCAL SLASHES PDP-11 FAMILY-SOFTWARE COSTS

Portland, Oregon — Now, Interactive Technology, Inc., has introduced Parallel Pascal, a complete, standard Pascal with extensions that dramatically reduces the cost of software support for DEC's new Falcon SBC-11/21 single-board computer, as well as for other PDP-11 processors with the RT-11 operating system. Parallel Pascal is priced at $950, compared with $8,500 for DEC's MicroPower/Pascal.

Versions of Parallel Pascal are planned for RSX operating systems and for other microprocessors.

For further information: Peter Mackie, President, ITI, Bob Anundson, V.P. Marketing, ITI, Interactive Technology Inc., 1225 NW Murray Road, Suite 103, Portland, Oregon 97229; (503) 644-0111.

February, 1982

EPS ANNOUNCES UNIX-COMPATIBLE MICRO VERSION OF FCS-EPS FINANCIAL PLANNING AND MODELING SOFTWARE SYSTEM AND INTRODUCES "THE DECISION SUPPORT MACHINE"

San Jose, CA — Robert M. Peak, Vice President of Sales for EPS, Inc. announced here today that the powerful FCS-EPS decision support system is now available for the ONYX Timesharing Super Microcomputer. EPS will sell the software by itself or with the microcomputer hardware as a "decision support machine."

FCS-EPS is a comprehensive software system for decision support using financial modeling, "what-if" scenarios, pre-written functions and routines, simultaneous equation solution, non-procedural statements, text manipulation, indirect addressing of variables, data management, forecasting, editing and advanced programming capabilities using the FCS-EPS language. A host of pre-written functions exist for depreciation, loans, NPV, lead and lag of payments/receipts, rounding, column calculations, percentages, summaries, etc.

Additional modules may be integrated for color graphics, hierarchical data management and consolidation, and a relational database manager facility.

Using FCS-EPS operating under the UNIX-compatible operating system on the ONYX Super Micro, the "Decision Support Machine" may have up to eight users on the same machine.

UNIX-compatible FCS-EPS software system introductory price is $6,000. The cost of an ONYX Super Micro computer and one million bytes of memory usually ranges from $27,500 - 36,650; however, the "Decision Support Machine" with FCS-EPS software, plus the ONYX with ½ MGB memory and an 18 MB disk and 4-user UNIX operating system may be purchased from EPS for $32,500.

For more information, contact, EPS, Inc., 1788 Technology Drive, San Jose, CA 95110, 800/538-7578 (in CA 408/292-6212).

March 8, 1982

EPS, INC. OPENS NEW SALES/SUPPORT OFFICE IN ATLANTA AND PHOENIX

Houston, TX — EPS, Inc. announces the opening of two new sales and application support offices to further strengthen usage of FCS-EPS, the computer based financial planning and data management system for accountants, planners, and analysts.

The new EPS offices are located at: EPS, Inc., P.O. Box 847, Atlanta, GA 30247, (404) 972-1980; EPS, Inc., P.O. Box 9128, Phoenix, AZ 85068, (602) 944-8906.

Today, FCS-EPS runs on more than 40 mainframe and mini computer systems in over 800 companies around the world. These installations are served by 13 other sales and support centers located in the U.S. and Canada, as well as numerous branch offices in Central and South America, Asia, Africa, Australia, Japan, Scandinavia, and Europe.

March 15, 1982

NEW RELEASE VERSION 3.0 VAX/VMS PERFORMANCE ANALYSIS "RABBIT-2" SOFTWARE

Atlanta, GA — Raxco Inc. announces the immediate release of Version 3.0 of RABBIT-2, a performance analysis software system for VAX/VMS environments.

RABBIT-2 is an interactive software tool that provides graphic representation of various system resources consummed by a single user, groups of users, projects, accounts, total system usage, or program images. Version 3.0 provides new graphic capabilities by incorporating advanced video features of VT100 terminals or lookalikes.

Other new features of Version 3.0 include automatic scaling for vertical and horizontal bar graphics, reverse image commands and bar selections. RABBIT-2 will now superimpose multiple graphs on the same display for comparison purposes. System data may be analyzed over any time period (e.g. Monday-Friday) and any interval of time (e.g. daily, hourly, minute by minute).

RABBIT-2 may be utilized by the system manager to investigate system bottlenecks, resource demand, user activities, and program utilization and analysis, It may be used interactively through a series of English-like commands, or via a batch file. Graphic output may be directed to the terminal or line printer.

Operational management may use RABBIT-2 as a planning tool to project future system requirements. The resulting graphs are easily

understood by all levels of management. RABBIT-2 provides system security features also, by dynamically reporting who was on the system on specified dates and times.

RABBIT-2 is alos currently available for RSTS/E users. An RSX11-M Plus version will be available in the second quarter of 1982.

RABBIT-2 is priced at $2495 for RSTS/E and $3995 for VMS systems. Rentals are $99/month and $200 month respectively. RABBIT-2 is sold and supported throughout the U.S.A., Canada, Europe and Australia.

RABBIT-2 is one of several integrated software systems provided by RAXCO Inc. Other VMS products include RABBIT-1 Resource Accounting and Billing and RABBIT-5 Incremental File Save and Restore. RSTS products include RABBIT-3 Job Accounting and Monitor and RABBIT-4 File Security. Data Management and Financial Planning Systems are also available for both systems.

For more information contact:
IN THE USA: RAXCO Inc., Suite 200, 6520 Powers Ferry Road, Atlanta, GA 30339, Telephone: (404) 955-2553; TWX: 810-766-2256; TELEX: 54-2659.
IN CANADA: RAXCO SOFTWARE Ltd., 18 Dowdy Street, Kingston, Ontario K7K 3V7, Telephone: (613) 549-7502.
IN EUROPE: HAMILTON RENTALS Ltd., Hamilton House, North Circle Road, London, England NW10 7UB, Telephone: 01-961-6777.
IN AUSTRALIA: DIGITAL EQUIPMENT AUSTRALIA Pty Ltd., Chatswood Plaza, Railway Street, Chatswood, NSW 2067 Australia, Telephone: 412 5252; Telex: 22988.

February, 1982
NEW RELEASE 3.0 "RABBIT-3" JOB ACCOUNTING and PERFORMANCE MONITOR for DEC RSTS/E USERS
West Palm Beach, FL — RAXCO Inc. announces the immediate availability of release 3.0 of RABBIT-3, Job Accounting and Performance Monitoring Software System for Digital Equipment Corp. RSTS/E operating system users.

The purpose of RABBIT-3 is to monitor system activities and create data that can be utilized by performance analysis (RABBIT-2) and/or billing (RABBIT-1) programs. RABBIT-3 output may also be utilized by Datatrieve, as well as Fortran, Cobol and Basic Users.

RABBIT-3 is a stand-alone system written in PDP macro that runs in 6-7 K memory. System degredation is minimal.

RABBIT-3 requires no sysgen and contains an auto-load parameter.

RABBIT-3 creates records for the following resource types:

Job Records . . . contain systems resources utilized by job.

CPU Statistics Record . . . contain periodic statistics in percentages of CPU utilization.

Disk Space Records . . . periodic records containing the disk space available.

Disk Catalog Record . . . information by filename, for all files on system including public and private disks.

Disk Statistics Records . . . contain statistical utilization information for each disk.

RABBIT-3 basic price is $2495 plus options. Rental is $99 per month plus options.

RABBIT Systems are in world-wide use on VAX-VMS and PDP-11 RSTS computers.

For more information contact: RAXCO Inc., 3336 N. Flagler Dr., West Palm Beach, Florida 33407. Tel: (305) 842-2115.

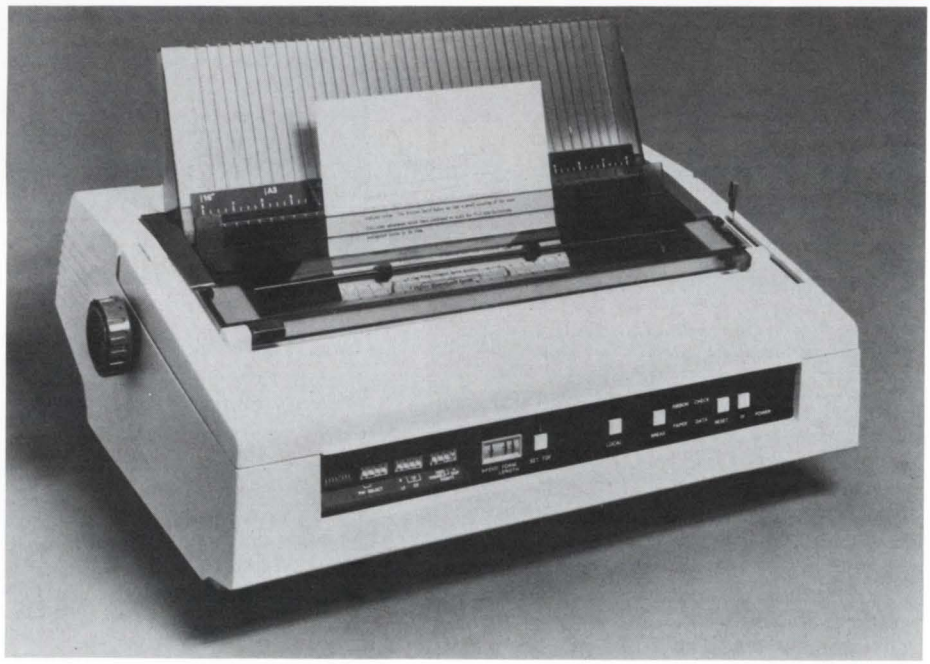
March, 1982
SOUTHERN SYSTEMS ENTERS LETTER-QUALITY PRINTER MARKET
Fort Lauderdale, FL — Southern Systems Inc. (SSI), leading add-on printer supplier for mainframes and mini-computers, has announced its entry into the letter-quality printer system market.

The new SSI Model 3500 Series prints at 33 characters per second (cps) while the SSI Model 7700 Series provides a 55 cps speed. Both are micro processor-controlled and offer features and options needed by word processor users.

Unmatched reliability results from the printers' 3000-hour MTBF, the highest in the industry, and a font life expectancy of 30 million impressions.

Interface-compatible with DEC, Data General, Hewlett Packard, Perkin Elmer, Burroughs, Modular Computer, Honeywell, TI and others, the two new letter-quality printer systems offer the lowest cost of ownership on the market, with single-board electronics and no requirements for adjustment or lubrication. The two models were created for the "office of the future" resulting in design emphasis on quietness.

Parallel or serial interfacing (synchronous and asynchronous) is offered. Standard word processing enhancements include automatic proportional spacing and built-in self-test.



Southern Systems Inc. (SSI) has introduced this Model 3500 Series, a 33 cps letter quality printer. The Fort Lauderdale, FL firm also announced a 55 cps letter quality printer, the Model 7700 Series.

Standard features also include advanced LSI design, one-piece universal power supply, digital positioning system, and minimal spare parts requirements.

Form length select switch is standard and a full range of forms-handling options is available including bi-directional tractor, cut sheet guide with auto load, front inserter, single bin cut sheet feeder, dual bin adapter and demand document tractor.

Cartridge ribbons allow for clean handling.

The new letter quality printer systems will be sold and serviced through Southern Systems' nationwide networks of sales and service representatives.

Both models consolidate logic electronics on a single board and offer a fast digital positioning system. The sophisticated paper control system uses a dual pressure roller assembly and a three-roller bail assembly that holds the paper firmly to the platen when friction feed is engaged. With this system, users gain smooth paper feeding without skewing and precise horizontal and vertical (line-to-line) registration of printed text.

On the Model 3500, word processing users have as standard a program mode that allows use of special characters and a graphics mode. These are available as options on the Model 7700.

Additional word processing options for both models include proportional space printing, half line feed, negative half line feed, automatic bold printing, shadow printing and underscore offset selection.

Southern Systems Inc. is located at 2841 Cypress Creek Road, Fort Lauderdale, FL 33309, (305) 979-1000.

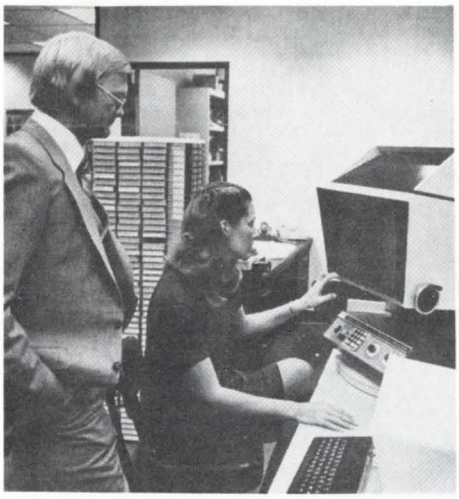
April, 1982
SATURN ANNOUNCES VIRTUAL UNIT FOR DEC USERS
Minneapolis, MI — By using Saturn Systems new Virtual Unit (VU) system, DEC users can divide a physical disk into many subfile directories using the RT-11 Disk structure. Up to eight units per user may be active at a time. VU is compatible with RT-11, TSX, and TSX-PLUS, and can be installed easily and quickly. Once a disk file is assigned to a virtual unit, it is treated as any other random access device. This enhances overall disk organization and segregation of programs and speeds directory accesses because there are fewer files in any one device directory.

Multiple files with the same file name can reside in a different VU on the same physical device and backup of specific data is greatly simplified. (If a copy of all data for a particular file is required, the entire contents of a VU can be copied.)

A number of security features are available with Saturn Systems VU system, including an optional password for assigning data to a specific VU. Also , when data is assigned to a VU, later access can be either read/write, read only, exclusive read/write, or exclusive write, depending on the specifications.

Saturn Systems, Inc. , is a Twin Cities firm specializing in software systems for use with the DEC family of minicomputers. For further information regarding the new VUW system, contact the company at 6875 Washington Avenue South, Suite 218, Minneapolis, Minnesota 55343. Phone: 800-328-6145; (612) 944-2452.

June, 1982
MICROGRAPHICS MANAGEMENT SYSTEM FOR RSTS/E
North Miami, FL — MMS, a Micrographics Management System that interfaces, indexes and automatically retrieves both microfiche, operture cards and microfilm is now available.



Designed to interface directly to 3M and Kodak Microfilm reader printers and provides automated storage and retrieval. The software is available both for PDP-11 RSTS/E and VAX/VMS computers.

The software features an easy to use Query language that includes a "Hit Count" feature as well as the software to interface VT-100's to the 3M and Kodak units.

Engineering documents on operture cards can also be indexed and retrieved automatically. Systems are installed in both the Federal Government and Fortune 500 companies.

For more information call or write, Florida Computer, Inc., John H. Wright, 99 NW 183rd St., North Miami, FL 33169, 305-652-1710 or in Europe contact Turnkey Software, 12 High Street, Chalfont St. Gile, Bucks, Canada HP8 4QA.

May, 1982

NEW APPROACH TO COMPUTER SECURITY

Bedford, MA — Clyde Digital Systems, a division of Clyde Enterprises and a principal supplier of application software, announces a stand alone software package for computer security. This software is implemented for the Digital Equipment Corporation PDP 11 series of computers, under the RSTS/E operation system. The approach is totally new, remarkably inexpensive and particularly effective. It permits security management personnel to monitor all interaction at randomly selected terminals at any time without interruption or intrusion into their job sessions. This monitoring activity cannot be detected by the user. All keystrokes entered at the user terminal and all information presented to the user terminal is also presented to the terminal of the security manager. In addition, the user's session dialogue is routed to a secured log file for subsequent review and formal documentation of covert activity.

Clyde Digital Systems has also implemented this capability in a fully automated mode that runs day and night without requiring the time and attention of security management personnel. In this mode it randomly selects, monitors, and records user session dialogues. With this security approach users know that their work at a terminal may be monitored and recorded at any time, and this without their being able to tell when such monitoring is taking place. This security application package is called CONTRL. It is available for immediate delivery. Priced at $595.00.

Product information may be obtained from; Janet (617) 275-6642.

May, 1982

FOX BOX MESSAGE GENERATOR (FB)

Pittsburgh, PA — FOX BOX is a hand held, battery operated message generator for asynchronous RS232 devices. U to 16 messages, each from 1 to 128 characters long, can be preprogrammed into the Fox Box EPROM. Character formats of 5 to 8 bits, one or 2 stops bits, even or odd parity may be intermixed through the 16 messages.

Take FB along on all trouble calls. Call up a Fox Box message for the terminal or device you are testing, at most any baud rate (14 to choose from).

Configure Fox Box to look like a model or as a terminal. All options are switch selectable, at your finger tip.

Standard PROM programs are available or use our instructions to provide your own custom programs.

SPECIFICATIONS:

Power — Battery powered, supplied with AC recharger
Size — 7.75" L x 4.75" W x 1.25" H
Weight — 1.5 pounds
Enclosure — Metal
Interface — RS232/V.24
Connector — DB25P, male, ribbon cable attached
Word Structure — 5-8 Data Bits; 1 or 2 stop bits; odd, even or no parity; all switch selectable
Baud Rate — 50-9600 baud, Switch Selectable
Message Storage — 2716 EPROM - Programming instructions included
Messages — 16 messages switch selectable
Message Length — 1-128 characters
Controls — Power on/off; Start Transmission
LED's — Transmit or Receive Data; Data Set Ready or Data Terminal Ready; Carrier Detect; Clear to Send
Configuration — Switch selectable between DTE & DCE
Delivery — Stock to 3 weeks
Black Box Price: Fox Box — (TSW30) — $445.00
Case — (TSX30) — N/C
Recharger — (TSY30) — N/C

Black Box Catalog, Mayview Road at Park Drive, P.O. Box 12800, Pittsburgh, PA 15241; 412-746-2910; TWX 510-697-3125.

May, 1982

INTERACTIVE MANAGEMENT SYSTEMS RELEASES "FIXED ASSET ACCOUNTING SYSTEM"

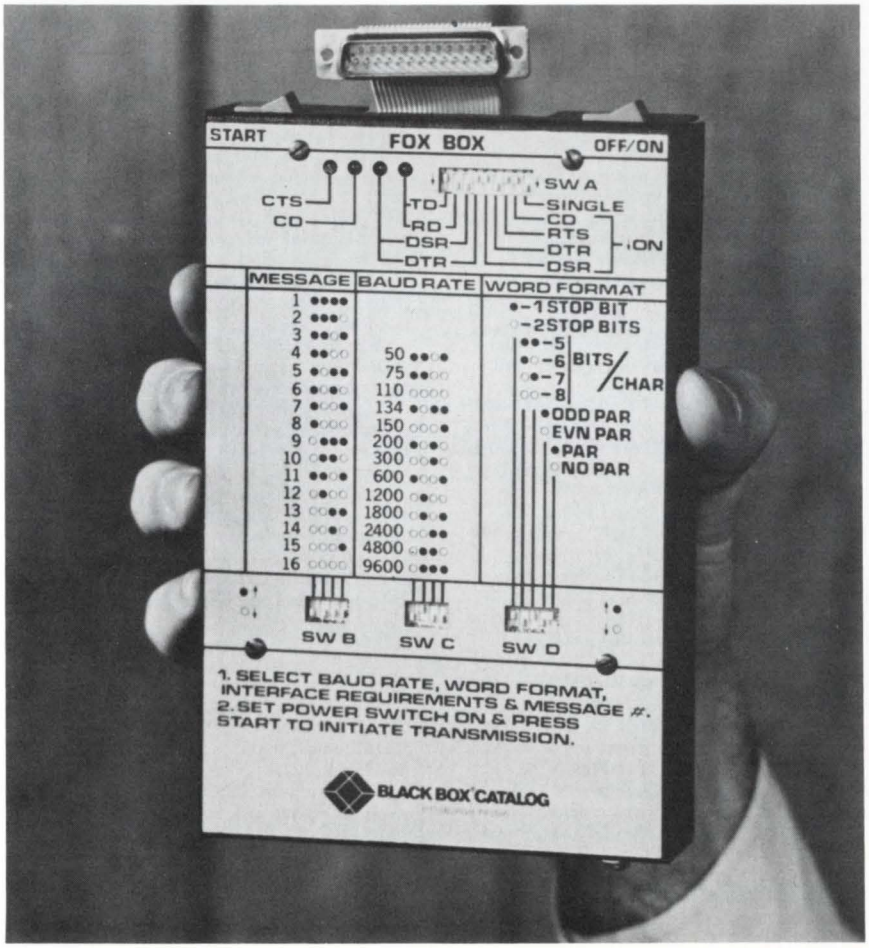
Belmont, MA — "Fixed Asset Accounting System", a new software package for Digital Equipment Corporation RSTS/E and VAX computers has recently been released by Interactive Management Systems. This new system was created to keep track of all depreciable and non-depreciable assets. This state-of-the-art product has been designed to meet the fixed asset requiremnts, as well as to conform to current accounting practices, of a wide variety of companies.

Totally up-to-date with current tax legislation, including the Economic Recovery Tax Act of 1981, this software ensures the maximum number of options in assigning depreciation methods. The IMS "Fixed Asset Accounting System" processes eight different depreciation methods. Since the system maintains separate financial and tax books, and two methods can be used for each asset.

The system supports the following depreciation methods: Sum of the years' digits method; 125% declining balance; 150% declining balance; 200% declining balance; unit of production method; straight line depreciation. In addition, the Economic Recovery Tax Act of 1981 has defined two depreciation methods of the Accelerated Cost Recovery System (A.C.R.S.) for tax reporting. The IMS "Fixed Asset Accounting System" covers both the standard A.C.R.S. method and the optional Straight-line A.C.R.S. method.

IMS also ensures that accepted accounting priciples are strictly observed and detailed reports guarantee complete audit trails. At the same time, the system is forgiving and user-friendly.

The IMS "Fixed Asset Accounting System" was designed especially to operate under Dig-



ital Equipment Corporation's VAX and RSTS/E family of computers.

"Fixed Assets" is only one of twelve software packages in the IMS fully integrated accounting and manufacturing product line, including: General Ledger; Accounts Payable; Accounts Receivable; Financial Planning; Payroll; Inventory Management and Control; Bill of Materials; Shop Floor Control/Job Costing; Material Requirements Planning; Purchasing; and Order Processing and Sales Analysis.

These software products have been installed in a broad range of companies world-wide. IMS clients range from electronic component manufacturers to national publications, from heavy equipment dealers to specialty job shops.

Since its founding in 1974, IMS has been a pioneer in the design, development, and implementation of applications software for Digital Equipment Corporation.

IMS was one of the first companies to develop applications software packages for DEC's RSTS/E, having worked on that machine since its inception. They have also been the leader in developing software for VAX.

IMS is continually upgrading and improving their products to keep current with Digital's latest technology. A high proportion of their revenues are funneled directly back into research and development. At the IMS Corporate Headquarters in Belmont, Massachusetts, there is an extensive computer center dedicated to R&D, documentation, and testing.

In addition to the company's own research, its geographical proximity to Digital's Corporate Headquarters allows them to serve as a "test-site" for new DEC products. Indeed, IMS strives to be among the first to learn about the latest changes in DEC's hardware, operating systems, and languages.

Interactive Management Systems prides itself on providing its clients with fully integrated state-of-the-art software and total professional service that reflects and changes with client needs and the most current technology.

The IMS Fixed Asset Accounting System simplifies the Implementation of the changes required by the Economic Recovery Tax Act of 1981.

May, 1982

WIREWORKS FORMS DATA COMMUNICATIONS DIVISION

Hillside, New Jersey — Wireworks Corporation, leading manufacturer of professional audio interconnect systems and broadcast quality video support systems, announces the formation of the Wireworks Data Communications Division. The new Division will focus o the data processing/computer community with a full line of interconnect cable systems and components. Previously, data/computer cable assemblies were manufactrued and distributed through Wireworks' Professional Audio Products and Broadcast Interconnect Divisions.

This initial product introduction includes an extensive line of RS-232C data cables, modified RS-232C cables, as well as coaxial and twinaxial cables — stocked in a variety of standard lengths for immediate delivery.

- DC Series cables are standard twenty-five conductor and modified RS-232C compatible assemblies, stocked in male/male, male/female and female/female connector configurations. Four, none and fifteen conductor cables (modified RS-232C) are money saving entries in this series since many computer systems do not require twenty-five line cables.
- EC Series cables are also RS-232C type calbe assemblies, but are manufactured utilizing technologically new, very low capacitance twisted pair cable, which also incorporates foil shielding. This low capacitance cable allows for longer cable runs than those achieved with conventioal dielectric type cables. The twisted pair construction reduces cross-talk within the cable and the foil shield guards agains EMI and RF interference. EC Series cables, like DC, are also available in full twenty-five line RS-232C and modified four, none and fifteen line RS-232C versions.
- BC Series cables are coaxial assemblies with BNC type connectors on both ends. These cables are utilized in IBM model 3270 and other similar systems. BC Series cables are stocked in a variety of standard lengths and in both male/male (i.e. equipment to equipment) and male/female (i.e. extension cable) configurations.
- TC Series cables are twinaxial assemblies, manufactured for use with IBM systems 34 and 38. TC Series cables provide the interconnect between IBM display station 5251 and the computer. Six stocked lengths are available for immediate delivery.

All cable types and styles may be ordered in any length required. Other connector cofigurations and custom wiring are also available.

All Wireworks cable assemblies are manufactured using the highest quality materials and controlled production and assembly procedures. Wireworks precision data communication cable assemblies are all individually packaged and ready for use. Share the security and confidence enjoyed by all Wireworks' valued customers.

For information and pricing about Wireworks Data Communication Cables, contact: Angela DiCicco, Custome Service Manager, Wireworks Corp., Box 3600, 380 Hillside Ave., Hillside, NJ 07205, tel: (201) 686-7400, Twx: 710-985-4675.

May, 1982

EMULEX INTRODUCES SC750 DISK CONTROLLER FOR USE WITH VAX-11/750 COMPUTERS

Santa Ana, CA — Emulex Corporation has introduced its SC750 disk controller designed exclusively for use on Digital Equipment Corporation's VAX-11/750 computer.

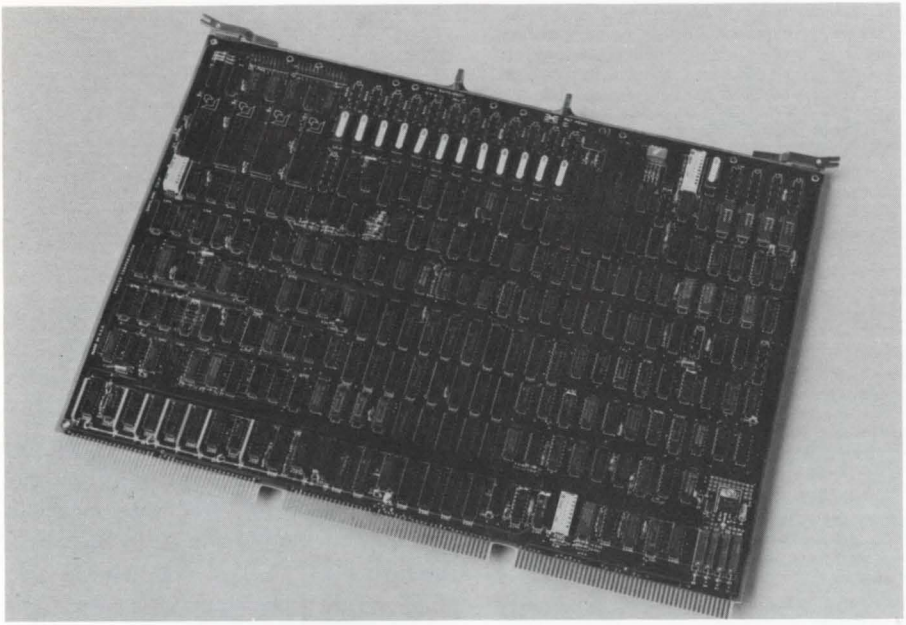
The new controller is functionally identical to the DEC RH750 Massbus adapter combined with RM03, RM05, and/or RM80 disks, making it possible for users to tie a wide variety of disk drives having SMD interfaces to the VAX-11/750. Since the controller emulates DEC Massbus disk subsystems, it is transparent to the DEC VMS and other VAS operating systems and to DEC diagnostics.

The controller is constructed on a single extended hexsize printed circuit board which mounts in any one of the three RH750 Massbus adapter slots in the VAX-11/750, without the need for wiring modifications. The controller interfaces directly with the 32-bit CPU Memory Interchange (CMI) bus of the VAX-11/750. its four SMD disk interfaces provide for direct connection to the disk drives without the need for additional "boat anchor" boxes of logic as is needed for DEC and other controllers. The controller contains the memory mapping and registers of the RH750 as well as all the DCL logic for up to four physical and eight logical drives.

The new disk controller is list priced at $8,950 in single quantities. Standard mix and match pricing, combining all Emulex disk, tape and communication controllers, may be applied to obtain volume OEM discounts. As an example, the purchase price of 50 or moe combined controller types would reduce the SC750 price to $6,086 per unit. The SC750 is available for delivery now.

"The SC750 uses the same bipolar microprocessor architecture as Emulex's SC21 Unibus disk controller," said Phillip (Flip) Begich, Director of National Marketing. "Users get all the same benefits, such as automatic self-test during power-up, intelligent buffer management, mixed drive handling, and reduced size, power and cost as with all other Emulex Controller products. In addition, the controller can operate at disk rates up to 15 MHz making it suitable for newer higher density disk drives such as the Fujitsu Model 2351.

This controller is the first of a series of new products designed exclusively for the VAS-11



The Emulex SC750 Series, the industries first large disk controller for DEC VAX-11/750 computers.

Designated the Model DQ204, the self-contained quad size controller requires only a flat ribbon cable and a single slot in any LSI based quad backplane for interconnection.

In operation the DQ204 offers a switch selectable choice of RL01 or RL02 software emulations, for four logical units, and is compatible with RT-11 and RSX-11 operating systems.

Other features offered by the controller include * automatic media-flaw compensation and retry on read errors * software write protect capability * automatic power down protection * on-board bootstrap loader for RL01/RL02 and TM-11 support with jumper selectable bootstrap address * automatic self-test feature * full sector data buffer for elimination of data-late errors due to DMA latency * memory addressing capability to 128K words * low power volts.

Price: $1,622.00 in quantity; Delivery: Stock to 30 days A.R.O.

Direct inquiries to: Corporate Headquarters—Mr. Dennis Edwards, Natl. Sls. Mgr., DILOG (Distributed Logic Corp.), 12800 Garden Grove Blvd., Garden Grove, CA 92643, Phone: (714) 534-8950; International Sales/Service Office—DILOG (Distributed Logic Corp.), 12 Temple Square, Aylesbury, Buckinghamshire, England, Phone: 44-296-34319 or 34310, Telex: 837 038.

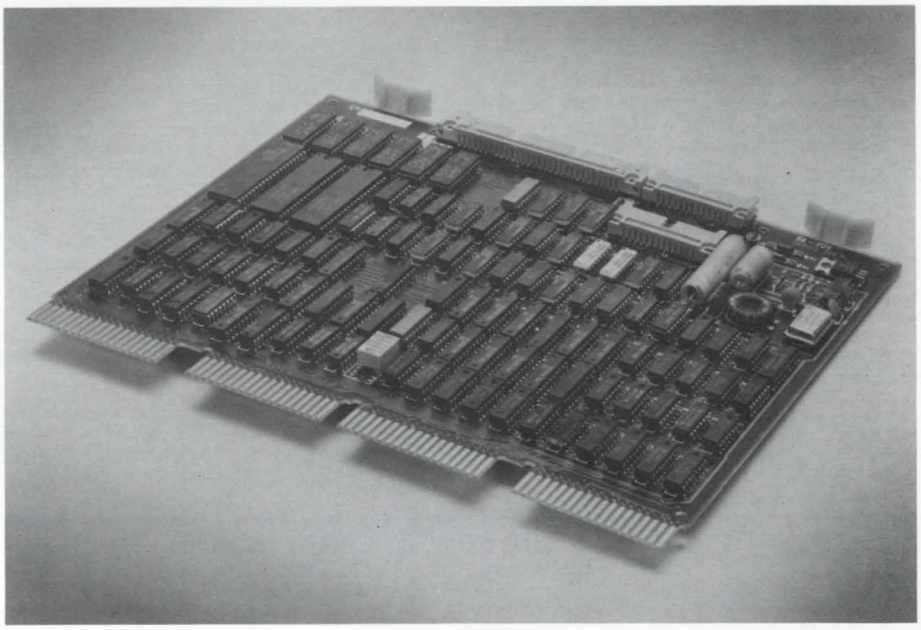
---

computers," Begich added. "We now and will continue to support our full range of Unibus disk and tape controllers on the VAX series since these products offer a very economical approach with acceptable performance in many applications. The SC750 now gives users complete flexibility in choosing the product that's best for their application."

Emulex Corporation, based in Santa Ana, is the leading supplier of disk, tape and communication controllers for use in interfacing a wide variety of peripheral devices to computers made by Digital Equipment Corporation. The company's new Systems Group also sells and installs complete disk subsystems ranging in size from 80 to 675 megabytes, for VAX and PDP-11 users, with nationwide service provided by Control Data Corporation/Engineering Services.

May, 1982
THE FIRST FULLY COMPATIBLE DISC CONTROLLER TO INTERFACE CDC LARK DRIVES TO DEC LSI-11 COMPUTERS OFFERS DEC SOFTWARE AND OPERATING SYSTEMS COMPATIBILITY

Garden Grove, CA — A new intelligent uP Disc Controller, which is the first to interface one or two current CDC LARK 8MB fixed and 8MB removable media drives is now available from DILOG (Distributed Logic Corp.), for interface with DEC LSI-11, 11/2 and 11/23 computers.



DILOG (Distributed Logic Corp.) introduces the first Disc Controller offering interface compatibility between CDC Lark drives and DEC LSI-11 computers.

PHOTO CONTEST

What has been TECO TESTED for QUALITY?

This was submitted by Peter Dick of Silver Programs, 267 Goldhawk Road, London W12 8EU, England.

A tee-shirt to someone with the correct answer.
[Of course, those connected with Silver Programs are not eligible to win.]

CORRECTION!

The following article appeared in the April, 1982 issue of **RSTS PROFESSIONAL**, (v.4, #2, p.47). Author Brad Smith (Allied Data, Olympia, Washington) sent us this note:

> ". . . I would like to correct a typographical omission. The last paragraph refers to "the use of parentheses in the next-to-last expression", but the next-to-last expression is missing. It should be
>
> 30% / 2% + (3% - 2%) = 15% + 1%.
>
> Needless to say, this omission will confuse readers. I hope that it will be rectified in a future issue...."

To lessen confusion, we have reproduced the entire article with the omitted line now in place and indicated with an arrow.

MORE NOTES ON LITERALS AND STRINGS IN BASIC-PLUS-2

By Brad Smith, Allied Data, Olympia, WA

The author has worked on PDP-11's for 5 years in several languages. He now specializes in the design and optimization of Basic-Plus-2 application systems.

In a previous article (RSTS Professional, December 1981), I explained the basic ways in which space for literals is allocated in Basic-Plus-2. Here is some additional information on ways to reduce the space and time required by a BP2 program.

One feature of the BP2 compiler which can be of importance is that concatenation of string literals is done at compile time. For instance,

```
A$ = "A" + "B"
```

produces the same object code as

```
A$ = "AB".
```

In addition, CHR$ functions with literal arguments are treated as literals: they are evaluated at compile time and can be concatenated with other literals at that time. This can help significantly in reducing the space and time required for printing. To use a simple example,

```
PRINT CHR$ (13%); CHR$ (10%);
```

requires 11 words to store the instructions plus a total of 12 bytes for the two literals. Concatenating them,

```
PRINT CHR$ (13%) + CHR$ (10%);
```

reduces the instruction space to 7 words and the data space to 6 bytes, and also reduces the execution time. Another example of the ways in which this compile-time concatenation can be utilized is in a keyboard input subroutine which returns a different value depending on the delimiter entered by the user. This can be done by writing something like

```
F% = POS(CR + LF + CHR$(27%) + CHR$(4%) + FF, D$, 1%)
```

where D$ is the delimiter entered by the user. Being aware of this feature enables the programmer to avoid the "expense" of storing the individual characters as elements in an array or concatenating the characters and storing the result in a variable to be used in the above expression — neither of those approaches is as efficient.

The evaluation of literal expressions applies also to numeric expressions, but only to a limited extent. The compiler has problems with the precedence of operators. In such a case, it will go as far as it can in simplifying the expression. Consider the following examples of integer expressions and how they are expressed in object code:

```
5%*6% / 2%                  = 15%
30% / (2% + 3%)             =  6%
5%*6% / (2% + 3%)           = 30% / 5%
30% / 2% + 3%-2%            = 16%
30% / 2% + (3% - 2%)        = 15% + 1%   <----
15% + (3%-2%)               = 16%
```

Enclosing 5%*6% in parentheses has no effect; however, note that the use of parentheses in the next-to-last expression, although not affecting the run-time result of the expression, does increase the space required to store it and the time to evaluate it.

One use of this feature is to make a program more readable — instead of

```
I% AND 53%
```

a programmer can write

```
I% AND (1% + 4% + 16% + 32%)
```

or `I% AND (1% OR 4% OR 16% OR 32%)`.

Lest you hope for too much from the compiler, I should point out that the SPACE$, STRING$, ASCII, LEN, LEFT$, RIGHT$, MID$ and SEG$ functions with literal arguments are evaluated at run time.

A couple of further notes on strings:

In order to examine the first character of a string, it is slightly faster (at least, on an 11/70 with FPP — this could vary in a different environment) and occupies the same space to say

```
CHR$ (ASCII(A1))
```

instead of

```
LEFT$ (A$,1%).
```

When comparing the first character of a string to a literal, it occupies less space (7 words instead of 8) to say

```
IF ASCII(A$) = ASCII ("A")
```

instead of

```
IF LEFT$(A1,1%) = "A".
```

It is still better (only 5 words of instructions, and 6 bytes less data due to the elimination of the string literal) to say

```
IF ASCII(A$) = 65%.
```

It may appear that use of these techniques could not effect significant results, but when you need another 5 miles per hour out of a program that appears to be at its speed limit, a few seemingly minor changes can add up to a noticeable improvement.

LIST OF ADVERTISERS

RSTS/E VERSION 7.1 DOES NOT REALIZE THAT THE BELOVED 'TS11' TAPE DRIVE EXISTS!!!

By Patrick Holmay
Computation Laboratory
St. John's University

Upon spending several hours backing-up both our RM03's, I prepared myself for what I assumed would be a long day generating version 7.1 of RSTS/E on our 11/70. Following the normal procedures described in the RSTS System's Generation Manual, I attempted to boot off the distribution media (magnetic tape) on the beloved TS11. I soon discovered that any attempt to bring up V7.1 from tape was fruitless. Each time I tried to boot off tape (hardware and software) the system would halt immediately.

After thoroughly frustrating myself, I called DEC asking for a solution to the problem. To my amazement, there were no software specialists to be found. Leaving a message with a receptionist, I finally received a phone call the following day. In my discussion with DEC, it was mentioned that the TS11 had not been configured into the new release of RSTS/E. Thus causing INIT to ignore the presence of this exceptional piece of equipment. (??????)

The answer to the problem: INIT needs a starting address to continue. Once the system halts, the operator's keyboard has to be set to console mode. In console mode a starting address of zero must be entered (in the form 'SO' <cr>). Upon doing this the system should continue on its merry way.

There lurks a possible disastrous element in all this; speaking from experience, the monitor that was installed and running on our 11/70 was somehow altered thus causing the system programs to do weird and wonderful things. My suggestion is to reinstall the monitor and bring the system up under normal conditions.

Be careful!!!

Also, users of TS11s may need to follow this procedure anytime they want to boot off of tape (version 7.1). DEC has not decided if they will patch this problem in this version of RSTS/E.

TIME & MONEY

Precious Commodities

WE'LL SAVE THEM BOTH!

systems northwest inc.

Introduces . . .

XDM '82

Extended Data Management For RSTS/E

TABLE DRIVEN	• Each XDM application stores system attributes for up to 255 systems and each XDM system stores up to 255 field descriptors. All table entries are easily maintained.
DB MAINT UTILITIES	• Add, change, delete and display data. Shrink, expand, examine and integrity check data and index files. Eases data base management.
MULTIPLE INDEXES	• Data can be accessed using up to 22 user-defined, dynamically maintained indexes per XDM system. Each application can be specifically tailored to meet the users need.
PROGRAMMER AIDS	• Aids package appends XDM routines to application programs, standardizes application programs, checks internal consistency, and formats field descriptors. A programmer's productivity is dramatically improved.
SUPPORT	• Includes phone consultation with SNW staff plus optional annual mag-tape update and on-site visitation.

SIMPLIFIES APPLICATION PROGRAMMING
IMPROVES PROGRAMMER COST EFFECTIVENESS
INCREASES PROGRAMMER PRODUCTIVITY

It's Your Time . . . It's Your Money . . . It's Your Choice! XDM '82

FOR MORE INFORMATION →

YOUR NAME _____
YOUR TITLE _____
COMPANY NAME _____
ADDRESS _____
_____
_____
_____
PHONE NO. _____

RETURN TO:

systems northwest inc.

114 SOUTH FRANKLIN STREET
SUITE 201
JUNEAU, ALASKA 99801

OR CALL
(907) 586-6140

CIRCLE 137 ON READER CARD

ABLE GOLD

ABLE IS STILL WINNING THE DECATHON

That rare symbol of achievement and stability in the wild world of DEC compatibility

ABLE is the leading independent supplier of DEC compatible products. No one else comes close. Probably no one else ever will come close because we are still charging along the path that put us on top in the first place. Today, we have the biggest selection in the industry and a pipeline full of new ideas for making your present system run better than ever. We deliver top performance at a competitive price, and we deliver on time. But there is one very important quality which sets us apart from the rest. We are the *innovators*. Fifteen out of our first eighteen products were industry "firsts" which gave DEC users like you their most significant way to achieve better performance without upgrading to a more expensive computer.

Every time we come in first, you come out ahead. That's why there are more than 5,000 ABLE clients worldwide involving 15,000 installations and $30,000,000 worth of products shipped to date. Our client list runs from "A" to "Z" and includes hundreds of the most famous names of all, along with the small and medium-sized companies which are just as important to us as their larger contemporaries. Find out about our complete line of UNIBUS-compatible general-purpose, special-memory and data-communications products as well as the MAGNUM™ series of computer systems. Find out about our deep commitment to post-sale service which goes far beyond delivering products and includes a worldwide customer support program long recognized as tops in the add-on market. Then go for the gold. Become an ABLE client.

ABLE'S FIFTEEN FAMOUS FIRSTS

1st UNIBUS Converter • UNIVERTER™

1st UNIBUS Cache • CACHE/45™

1st 8KByte UNIBUS Cache • CACHE/434™

1st Full 256KByte FASTBUS Memory Add-In • SCAT/45™

1st Dual-Width In-Line Bus Repeater • REBUS™

1st Single-Board 4-Line DL Family • QUADRASYNC™

1st Single-Board 4-Line Auto Dialer • QUADRACALL™

1st 19.2 KBaud DH Communications Multiplexer • DMAX/16™

1st Full-Capability DV Replacement • ABLE DV/16

1st Single-Board 16-Line DZ • ABLE DZ/16

1st Single-Board DH • ABLE DH/DM

1st Single-Board Data Communications Multiplexer with Up/Down Compatibility DZ to DH and More with Simple, Socketed, ROM Changes • ABLE Vax DZ

1st 11/34 RSTS Memory Expander Allowing Full 4MByte Addressability • ENABLE/34™

1st HiRel 11/24-11/44 Computer System Alternative, a Low Maintenance Industrial Product at a Fair Price • 34 & 44 MAGNUM™

1st 11/23 UNIBUS Converter with I/O Map Allowing Full 4MByte Addressability and Full RSTS and RSX Application • UNIMAP™

WHEN WE COME IN FIRST, YOU COME OUT AHEAD. ABLE PRODUCTS MAKE ANY UNIBUS SYSTEM RUN STRONGER AND LAST LONGER.

ABLE the computer experts

ABLE COMPUTER,
1732 Reynolds Avenue,
Irvine, California 97214.
(714) 979-7030.
TWX 910-595-1729 ACT IRIN.

ABLE COMPUTER, ABLE Computer House,
London Road, Newbury,
Berkshire, England RG13 2QJ.
44(0635) 32125.
TELEX 848715 ABLE G.

ABLE COMPUTER GmbH,
Forsthausstrasse 1, 8013 Haar
(Near Munich), West Germany.
49 089/463080, 463089.
TELEX 05213883 ABLE D.

CIRCLE 56 ON READER CARD

That rare symbol of achievement and stability in the wild world of DEC compatibility

ABLE is the leading independent supplier of DEC compatible products. No one else comes close. Probably no one else ever will come close because we are still charging along the path that put us on top in the first place. Today, we have the biggest selection in the industry and a pipeline full of new ideas for making your present system run better than ever. We deliver top performance at a competitive price, and we deliver on time. But there is one very important quality which sets us apart from the rest. We are the *innovators*. Fifteen out of our first eighteen products were industry "firsts" which gave DEC users like you their most significant way to achieve better performance without upgrading to a more expensive computer.

Every time we come in first, you come out ahead. That's why there are more than 5,000 ABLE clients worldwide involving 15,000 installations and $30,000,000 worth of products shipped to date. Our client list runs from "A" to "Z" and includes hundreds of the most famous names of all, along with the small and medium-sized companies which are just as important to us as their larger contemporaries. Find out about our complete line of UNIBUS-compatible general-purpose, special-memory and data-communications products as well as the MAGNUM™ series of computer systems. Find out about our deep commitment to post-sale service which goes far beyond delivering products and includes a worldwide customer support program long recognized as tops in the add-on market. Then go for the gold. Become an ABLE client.

ABLE'S FIFTEEN FAMOUS FIRSTS

1st UNIBUS Converter • UNIVERTER™

1st UNIBUS Cache • CACHE/45™

1st 8KByte UNIBUS Cache • CACHE/434™

1st Full 256KByte FASTBUS Memory Add-In • SCAT/45™

1st Dual-Width In-Line Bus Repeater • REBUS™

1st Single-Board 4-Line DL Family • QUADRASYNC™

1st Single-Board 4-Line Auto Dialer • QUADRACALL™

1st 19.2 KBaud DH Communications Multiplexer • DMAX/16™

1st Full-Capability DV Replacement • ABLE DV/16

1st Single-Board 16-Line DZ • ABLE DZ/16

1st Single-Board DH • ABLE DH/DM

1st Single-Board Data Communications Multiplexer with Up/Down Compatibility DZ to DH and More with Simple, Socketed, ROM Changes • ABLE VAX DZ

1st 11/34 RSTS Memory Expander Allowing Full 4MByte Addressability • ENABLE/34™

1st HiRel 11/24-11/44 Computer System Alternative, a Low Maintenance Industrial Product at a Fair Price • 34 & 44 MAGNUM™

1st 11/23 UNIBUS Converter with I/O Map Allowing Full 4MByte Addressability and Full RSTS and RSX Application • UNIMAP™

WHEN WE COME IN FIRST, YOU COME OUT AHEAD. ABLE PRODUCTS MAKE ANY UNIBUS SYSTEM RUN STRONGER AND LAST LONGER.

ABLE the computer experts

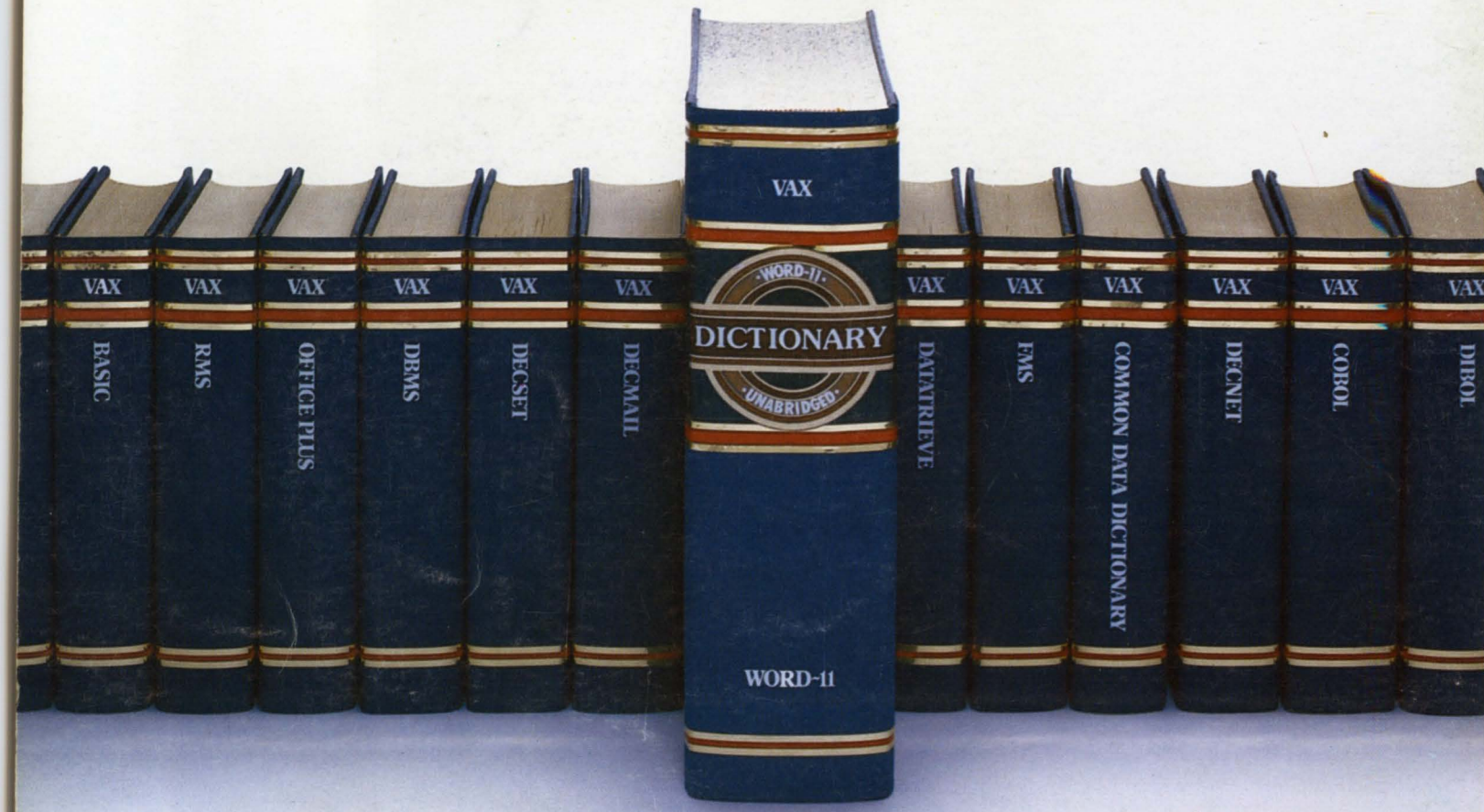
ABLE COMPUTER,
1732 Reynolds Avenue,
Irvine, California 97214.
(714) 979-7030.
TWX 910-595-1729 ACT IRIN.

ABLE COMPUTER, ABLE Computer House,
London Road, Newbury,
Berkshire, England RG13 2QJ.
44(0635) 32125.
TELEX 848715 ABLE G.

ABLE COMPUTER GmbH,
Forsthausstrasse 1, 8013 Haar
(Near Munich), West Germany.
49 089/463080, 463089.
TELEX 05213883 ABLE D.

CIRCLE 56 ON READER CARD

WORD-11.
The Word Processing System Vital For VAX.



The Unabridged Addition.

No matter how many additions you've made to your VAX,™ it won't be complete without WORD-11.™ It's the sophisticated word processing system designed to help you use all the other information you've stored. With features like list processing, built-in dictionaries for spelling error detection, automatic table of contents and footnoting that make report writing a snap.

WORD-11 runs concurrently with data processing on multiple terminals. It's easy to use. And it's been up and running in hundreds of installations all over the world for years.

WORD-11 is easy to install and operate. It's cost effective. And it's supported by a dedicated team of experienced specialists. It could be the most important addition you make to your VAX installation. For details, Call, (714) 993-4160.

WORD-11 is also available for Digital's RSX-11M,™ RSX-11M PLUS™ and RSTS/E™ operating systems.

Data Processing Design, Inc.
AUTHORIZED digital COMPUTER DISTRIBUTOR

CORPORATE OFFICE
181 W. Orangethorpe, Suite F
Placentia, CA 92670
714-993-4160 Telex 182-278

N Y OFFICE
420 Lexington Avenue, Suite 633
New York, NY 10170
212-687-0104

WASHINGTON D C OFFICE
4520 East-West Highway, Suite 550
Bethesda, MD 20814
301-657-4098

CIRCLE 59 ON READER CARD