THE mini-tasker

DECUS
RT-11 SIG NEWSLETTER

March 1985

Volume 11 Number 1

SJ RTMON RMON FILEX CSI ODT

PIP LD SYSMAC FB

DIR QUEMAN KMON

QUEUE TECO

PAT DUP

VM

KED BINCOM K52

LIBR DUMP BUP

SIPP LINK

SRCCOM RESORC FORMAT

SLP TTYSET IND

MACRO XM JSW HELP

RT-11 SIG DECUS U.S. CHAPTER

"The Following are trademarks of Digital Equipment Corporation"

ALL-IN-1
DEC
DECnet
DECmate
DECsystem-10
DECSYSTEM-20
DECUS
DECwriter
DIBOL
Digital logo
EduSystem
IAS
MASSBUS
PDP
PDT
P/OS
Professional
Rainbow
RSTS
RSX
RT
UNIBUS
VAX
VMS
VT
Work Processor

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation or DECUS. Digital Equipment Corporation and DECUS assume no responsibility for any errors that may appear in this document.

POLICY NOTICE TO ALL ATTENDEES OR CONTRIBUTORS "DECUS PRESENTATIONS, PUBLICATIONS, PROGRAMS, OR ANY OTHER PRODUCT WILL NOT CONTAIN TECHNICAL DATA/INFORMATION THAT IS PROPRIETARY, CLASSIFIED UNDER U.S. GOVERNED BY THE U.S. DEPARTMENT OF STATE'S INTERNATIONAL TRAFFIC IN ARMS REGULATIONS (ITAR)."

DECUS and Digital Equipment Corporation make no representation that in the interconnection of products in the manner described herein will not infringe on any existing or future patent rights nor do the descriptions contained herein imply the granting of licenses to utilize any software so described or to make, use or sell equipment constructed in accordance with these descriptions.

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsiblity of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility of liability for articles or information appearing in the document. The views herein expressed are those of the authors and do not necessarily express the views of DECUS or Digital Equipment Corporation.

THE mini-tasker

DECUS
RT-11 SIG NEWSLETTER

March 1985 Volume 11 Number 1

Contributions to the newsletter should be sent to:

Ken Demers
Adaptive Automation, Inc.
5 Science Park
New Haven, CT 06511
203 786-5050

Other communications can be sent to:

John T. Rasted
JTR Associates
58 Rasted Lane
Meriden, CT 06450
203 634-1632

or

RT-11 SIG
C/O DECUS
One Iron Way
MR2-3/E55
Marlboro, MA 01752
617 481-9511 Ext. 4141

Table of Contents

SIG CONTRACTS

RT-11 SIG Chairman
JOHN T. RASTED
JTR Associates
58 Rasted Lane
Meriden, CT 06450
(203) 634-1632

Newsletter Editor
DECNET Contact
KEN DEMERS
Adaptive Automation
5 Science Park
New Haven, CT 06511
(203) 786-5050

Business Working Group Contact
KENNETH L. AYDLOTT
Teledyne Hastings-Raydist
P.O. Box 1275
Hampton, VA 23661
(804) 723-6531

Tape Copy Generation Contact
RALSTON BARNARD
Division 7523
Sandia Laboratories
Albuquerque, NM 87185
(505) 844-5115

Structured Languages Contact
MARK BARTELT
HSC Research Development Corp.
555 University Ave.
Toronto, Ontario
Canada M5G 1X8
(416) 597-1500 Ext 4588
or 598-5982

APL Contact
DOUG BOHRER
Bohrer & Company
903 Ridge Rd
Suite 3
Wilmette, IL 60091
(312) 251-9449

TSX and MACRO Contact
NICK BOURGEOIS / 5238
Software Systems Group
3236 Candlelight N.E.
Albuquerque, NM 87111
(505) 298-2346

TECO Contact
Product Planning Contact
JOHN CROWELL
CROW4ELL Ltd.
145 Andanada
Los Alamos, NM 87544
(505) 662-3893

COBOL Contact
Wish List Contact
BILL LEROY
Resourceful Solutions
470 E. Paces Ferry Rd. N.E.
P.O. Box 52661
Atlanta, GA 30355
(404) 231-3117

RT-11 Hardware Contact
CARL LOWENSTEIN
Univ. of Calif., San Diego
Marine Physical Lab. of the
Scripps Inst. of Oceanography
San Diego. CA 92152
(619) 294-3678

Standards Coordinator
BRADFORD LUBELL
Univ. of Calif., Los Angeles
A3-381 CHS
Los Angeles, CA 90024
(213) 825-6713, 206-6119

C Contact
JACK PETERSON
Horizon Data Systems
1901 Wildflower Terrace
Richmond, VA 23233
(804) 740-9244

Communications Committee Rep.
FMS Contact
Susan Rasted
Software Dynamics Inc.
1000 Yale Ave.
P.O. Box 500
Wallingford, CT 06492
(203) 265-2226

Symposia Coordinator
NED RHODES
Software Systems Group
1684 East Gude Drive
Rockville, MD 20850
(301) 340-2773

RT DECUS Library Contact
Tape Copy Distribution
TOM SHINAL
General Scientific Corp.
1684 East Gude Drive
Rockville, MD 20850
(301) 340-2773

Pre-Symposium Seminar
Coordinator
RT SIG Suite Mgr.
Bruce D. Sidlinger
ALCOR Inc.
10130 Jones-Maltsberger Rd.
P.O. Box 32516
San Antonio, TX 78284
(512) 349-3771

BASIC Contact
ED STEVENS
EMDA Inc.
77 North Oak Knoll
Suite 104
Pasadena, CA 91101
(213) 795-5991

CAMAC Contact
J. W. TIPPIE
Kinetic Systems Inc.
11 Mary Knoll Drive
Lockport, IL 60441
(815) 838-0005

Personal Computer Contact
LUG Contact
WILLIAM K. WALKER
Monsanto Research Corporation
MRC - Mound
Miamisburg, OH 45342
(513) 865-3557

FORTRAN Contact
RUNOFF Contact
ROBERT WALRAVEN
Multiware, Inc.
139 G Street, Box 161
Davis, CA 95616
(916) 756-3291

USER INPUT

Recently I bought a model PR77EX EPROM programmer which is made by INTERPLEX INC. The EPROM burner software provided uses .SAV files to burn into the EPROMS. After experimenting with the board I found that there was a problem with the bitmap being written in the vector area. The manuals state that the bitmap is there to tell RT how big the program is. In my case, the bitmap was unnecessary, because RT was not going to run the program. In fact, the program would not run properly with the bitmap in block zero.

After another look at the manual I found the LINK/NOBITMAP command. Due to the size of the program LINK/NOBITMAP generated a "LINK-F-Storing text beyond high limit" error when this option was tried.

I looked at the source code for version 4 as a last resort. It seems that the NOBITMAP option does a lot more than inhibit the bitmap. It also generates code suitable to run on a RSTS machine. Whatever RSTS needs was causing LINK to go over the limit.

In the commented version 4 sources the section that actually writes the bitmap is rather easy to find (this shocked and amazed me). The patch included here will generate a version of LINK that doesn't write the bitmap at all. This is not meant to be a replacement for LINK, but a separate utility. It generates a program LINK2.SAV to be used to generate a pure but non-executable image. This patch works on version 5.01 RT-11 as distributed with Micropower Pascal (and possibly all LINKs 5.00 and above).

This is not supported by DEC, of course, but I would like to see a LINK/NOBITMAP:ONLY included in RT (hopefully before version 32).

```
COPY LINK.SAV LINK2.SAV
R SIPP
DK:LINK2.SAV/C
7
3462
4176
240
^Z
4204
240
^Z
4220
240
^Y
52427
^C
```

John T. Davies III
Thermo Electron Corporation
Instruments Division
524 Alpha Drive
Pittsburgh, PA 15238-2912
(412) 963-0907

WHY WON'T IT BOOT?

The following is adapted from a contribution by Mark Hansen, which in itself was the conclusion to a two week long puzzle. Mark has an 11/23 with 256KB of memory and some 3rd-party disks. He is running RT-11, but the story is probably of interest to any user of LSI-11s. The problem started when he decided to upgrade to V5.1B from V5.0. He put a copy of the distribution on the hard disk and went through the SYSGEN. Now Mark, like most other RT-11 users, is not a systems-guru: he is a user. Nonetheless, he read the SYSGEN manual and followed the instructions (which these days are very clear and simple), and all seemed to be well. Then he tried to boot the new version – and the system died. He checked what he had done but to no avail. The system just died immediately on booting. The old version, on the other hand, booted just fine. He came over to see me and we went through all the steps he had taken. No joy. Perhaps, I thought, the 3rd-party disks weren't all that good at emulating the DEC ones, and the system was waiting on a disk interrupt. Mark went back and checked in more detail just what the system was doing when it died. After much patient ODT he was able to report that the system was halting because the program space had been cleared to all zeroes, and that prior to this it had been in the middle of the main (disk-resident) bootstrap, in the section entitled "KT11 registers and enable memory mapping". So the disk drives were probably not to blame. Mark went back and thought about this (spelt "more ODT") and returned to report that the code seemed to be probing through the memory space with a loop counter, and that drastically reducing the loop counter would let the system boot. He also looked at the V5.0 code and found that it looked quite different. The "panel" sat down and brainstormed. It was noted that the reduction in the the loop counter corresponded roughly with the difference between 256kb and 4Mb, and that the new code had been significantly altered with lots of conditional sections depending on a flag which looked like "PRO". Question: how much memory did Mark have? Answer: an MXV11 with 8kb and an MSV11 with 256kb. Aha! Is the memory correctly jumpered to start at 8k? Mark thought so, although he couldn't be sure: the system had been built from bits and upgraded some time ago.

Then inspiration struck. A quick check of the code again showed that the new version was probing for memory in small steps and presumably clearing HIGH memory, but since the MXV11 was only an 18-bit address card, it was responding at 256+ and was being cleared while it still had the boot code in it! Reducing the loop counter prevented the system from probing above 256kb and so saved the MXV11. The solution was obvious: disable the MXV11 from responding to addresses over 18 bits. But HOW, especially as the MXV11 cannot have its memory disabled?

At this point we remembered a comment made by one of our good friends within the Tower, and looked more closely. (It's a bit like Dungeon: a clue here, a clue there.) The logic goes like this: the address decoder on the card will respond when the address is within range and the correct sync signals occur on the bus. The valid address will be latched and the memory will be accessed, either written or read. A reply will go on the bus (no reply gives a bus-timeout, remember?). But what if the address decoding logic chips have a spare enable input? If this enable input was changed, the board would no longer respond, would it? And there, in the middle of the MXV11 board, is a jumper. Actually, it was a spare board we were looking at, not Mark's. But back he went to his site and hauled the board out. Now taking a pair of cutters to a genuine DEC board is tantamount to sacriledge, but Mark is an RT-11 user (not one of the lesser breeds), so he cut it, moved the MSV11 memory down to start at address zero, where any good memory card should start, and the system booted! We conjecture that this bit of V5 was still designed for 18-bit addressing (there certainly were problems with the 22-bit addressing) and that V5.1 is for 22-bit addressing, including the PRO300 series. Why we had not heard of this problem earlier is uncertain, but perhaps not many people have installed V5.1B in Australia on such lab systems yet. Anyhow, problem solved.

For those who have MXV11 cards and would like to know what we did (and it is relevant to RT, RSX and RSTS), the following details may be of interest. First of all, the jumper is NOT documented in the standard DEC books, nor in the Field Service pocket books. Basically, the logic link between BSYNC and BRPLY has to be broken. W4 and W5 (4k and 16k links) cannot do this, and neither can the links in the range J30–J34. Put the card down in front of you, handle away and connector near. The addressing jumpers are in a big row up the middle of the card. Near the top of the jumpers and to the right there is a small block of about 5 horizontal resistors. Look closely and you will see two brown ones with a single black band each. This single black band means zero ohms. These are not resistors at all, they are fancy disguised wire links! In the block the nearest resistor (to the edge connector) is a real resistor, then there is a zero ohm link, and some other resistors. There may even be a cut link near the top. (One wonders what it used to do?) The second "resistor", the zero ohm one, is IT! We cut it and it worked.

CAVEAT

The above details are furnished for information only: no responsibility for the accuracy or otherwise can be accepted by DECUS or any of those mentioned. It is possible that the jumper mentioned may leave an open TTL gate input, subject to interference: this has not been checked. The system has been running with the cut link for only a little while, so long term reliability has not been checked. We would be very interested to hear from anyone else who has tried out this or anything similar (succesful or otherwise).

With thanks to Mark for the problem and the write-up, to Powell, Phil and Chris for brainstorming, and to John-in-the-Tower for the clue.

R N Caffin
CSIRO Division of Textile Physics

I presented a short description at Anaheim of how we read RT-11 tapes on our VAX. John Rasted wanted me to send our procedure for the SIG newsletter since most of RT-11 land does not go there. Also, since then I have had occasion to spend a little more time and thus have enhanced the method some.

Problems with Reading RT-11 Mag Tapes Under VMS

Many of us do not have mag tapes on our RT-11 systems. Once upon a time, we could copy the files from tapes onto an RT-11 volume under RSX-11. However, it is frequently necessary to use VMS since the RSX systems are all busy.

For some reason, VMS often gets confused about the file names on an RT-11 tape. It will append " at the front and back of the file name for instance. When you try to use the COPY command, VMS will reject all such names as being illegal.

One other problem that arises is the RMS appends carriage-return linefeeds every 512 bytes. It does not recognize RT-11 file structure.

This article describes our method for dealing with these problems.

Reading the Files

First, mount the tape as follows:

```
MOUNT/BLOCKSIZE=512/OVER=ID MTA0:
```

The block size qualifier is necessary since RT-11 tape volumes do not include HDR2 labels.

Now, get a directory of the tape:

```
DIR/COLUMN:1/OUT=MTA0.COM MTA0:
```

Print MTA0.COM since we are going to convert it into a rename command procedure.

Copy all files into your VMS directory as follows:

```
COPY MTA0:*.* A.A
```

The wildcard copy overcomes the file name problems. It copies the files into ascending versions of A.A.

Now edit the command file to start each line with:

```
$RENAME A.A;0 <filename>
```

Eliminate the extraneous characters from the filenames as necessary.

Converting the Files for Transfer to RT-11

The VMS files will contain CR-LF every 512 bytes. These can be eliminated under VMS by these commands: (you will probably want to execute these via a command file if there are many files)

```
ANALYZE/RMS_FILE/FDL/OUT=T.FDL <filename>
```

Edit T.FDL to change the carriage control value from carriage_return to none.

```
CONVERT/FDL=T.FDL <oldfilename> <newfilename>
```

This command removes the extra CR-LF bytes.

The result of these commands are files which you can move to an RT-11 disk volume via FLX.

Converting the Files for Use Under VMS

EDT will still believe that the files contain 512 byte records. If you need to edit or use the file under VMS, use TECO as follows:

```
TECO <outfile>=<infile>

*EX$$
```

The result of this action will be an RMS file with the correct attributes. There is probably a better way to convert the file, but we have not been able to find it quickly.

Regards,

Jim Norris

Jim Norris
Mail Zone 1174
(817) 777-2617

GENERAL DYNAMICS
Data Systems Division
Central Center
P.O. Box 748, Fort Worth, Tx 76101
•817 732-4811 •817 731-0741

From: Kenneth L Aydlott

TO: RTSIG DISTRIBUTION LIST (@RTSIG)

Return-Receipt requested
Subject: SIMRT, LDAs and 11/23

We presently have a system that uses the 11/2. Our FORTRAN IV program is LINKed into .LDA format with SIMRT. The .LDA and ABSLDR are put on tape. This tape is put into the 11/2 system and the power is switched ON. The 11/2 is strapped into ODT mode so that it loads the ABSLDR from the tape. The ABSLDR, in turn loads the .LDA file. The whole process is begun with the L command to the 11/2's ODT. This command causes the microcode BOOTSTRAP LOADER to load the ABSLDR.

As you all know, the 11/2 is no longer manufactured. We want to upgrade the system to 11/23. The problem is that the 11/23's ODT has no L command and, therefore, we assume, has no BOOTSTRAP LOADER.

1. HOW can we get these systems to load our .LDAs similar to the way it is now being done?

2. CAN we get the 11/23 systems to load the .LDAs in a similar fashion?

3. Is the MRV11C our only hope? (It is power hungry: 0.8A @5VDC.)

4. What about SIMRT under V5.01? Are there any known problems?

Answers/suggestions to any of the above questions are welcomed.

Thanx,
Ken

rom: Ned W. Rhodes
To: Kenneth L Aydlott (AYDLOTT)
CC: RTSIG DISTRIBUTION LIST (@RTSIG)

Subject: In Reply To SIMRT, LDAs and 11/23

Suggest that you make your own boot prom and use that to load in your new stuff.

Have you explored the possibility of a bubble memory disk for the system. You can get them in various sizes and it would allow you to run REAL RT-11 rather than having to go to all the problem associated with .LDA format. Other advantages would be that you could take advantage of the XM monitor for virtual overlays and other neat memory management things. Don't know the power requirements, but it would allow you to really use an 11/23 as a 23 rather than a fast 11/2.

I have no SIMRT experience, but how about doing a paper on it so that you can share your experiences with others. (Don't you just love these kinds of responses?). I'll expect a paper for the Fall in Anahiem.

From: Kenneth L Aydlott

TO: RTSIG DISTRIBUTION LIST (@RTSIG)

Return-Receipt requested
Subject: SIMRT, LDAs and 11/23 ANS

1) I found a listing of the BOOTSTRAP LOADER in the 1975-76 11/03 Handbook. I disassembled it which explained how the ABSLDR was loaded.

2) I did a byte dump of ABSLDR because the first byte is the BOOTSTRAP LOADER format address offset, thus offsetting all the instructions by one byte.

3) I SIPPed the ABSLDR in using the byte dump listing and verified a byte dump of this against the original byte dump.

4) I DUMPed this new ABSLDR and used this dump listing to make an ASCII file for input to microcode ODT.

5) I recompiled our old FORTRAN IV program using V2.6 compiler and reLINKed with the 11/23 library and $SIMRT.

6) We output the ABSLDR.ODT and FIELDB.LDA files to the 11/23 using the "toggled in" ABSLDR to load the .LDA file. SURPRISE! SURPRISE! the program runs indicating that SIMRT under V5.01B is ok. We have not checked everything out 100%, closer to about 85-90%. Needless to say, we are very optimistic that the whole thing will work without spending big $$ on bubble memory.

The nice thing is that this setup (step 6) works just fine on the 11/02 as well so that the customer doesn't have to upgrade the CPU and yet still get all the new, improved capabilities.

(There's your paper, Ned!)

Ken

P.S. I've written down the disassembly of the old BOOTSTRAP LOADER so that I need not repeat that "joyous" feat.

rom: William K. Walker
To: Ned W. Rhodes (RHODES)
CC: Kenneth L Aydlott (AYDLOTT)
CC: RTSIG DISTRIBUTION LIST (@RTSIG)
Return-Receipt requested

Subject: In Reply To SIMRT, LDAs and 11/23

Ned,

Bubbles are kind of expensive, though. As I recall, 256K is about $2500, plus the cost of the controller. Ah, if I could only afford to put about 5MB of bubble memory on my 11/73...

Bill

FORTRAN SUPPORT for .PEEK and .POKE EMTs

Here is a little tidbit that maybe some of the newsletter readers may find useful. We at the EMC (ElectoMagnetic Compatability) LAB need to do low memory peeks and pokes for some Fortran programs, as we run RT on an 11/20 (a PDP 11 WHAT???) and TSX+ on an 11/73.

Until RT v5.1 came along, we solved the problem of low-memory access by using a modified version of the DECUS XD handler. This gave us subdevices (.DEV) under RT/TSX and low-memory access. With RT V5, LD replaced XD for use with subdevices (.DSK), but we still needed XD for low-memory access.

While reading the new documentation for V5, we came across the answer for low-memory access - .PEEK and .POKE. Trying to find the description for these routines in SYSLIB, the manual stated: "IPEEK and IPOKE not equivalent to .PEEK and .POKE".

To solve our delema, we came up with a simple solution. Write two macro routines, callable by Fortran, to use .PEEK and .POKE. As such, we now have two low-memory access routines, that operate in the same manner, under all operating systems - RT (SJ,FB,XM) and TSX+.

The routines are called the same way as IPEEK and IPOKE, and return the same result. FUNCTION LMPEEK returnes the INTEGER*2 value contained in address IADDR, and SUBROUTINE LMPOKE inserts the INTEGER*2 value IVAL into address IADDR. The descriptions are the same as the SYSLIB routines IPEEK and IPOKE, and their operation is as described for .PEEK/.POKE programmed requests. Note that the operation under TSX is slightly different than under RT.

Simply MACRO the routines, and then optionally put them in a library of your choice. If there are any questions, give me a call or write. The numbers or given below.

Here are the routines:

```
.TITLE  LMPEEK             ;FORTRAN callable .PEEK EMT FUNCTION
.IDENT  /V01.00/           ; The first try
.ENABL  LC

.GLOBL  LMPEEK
.MCALL  .PEEK

OCT 18/84
MIKE MARAK
DAVID GAUDINE
EMC LAB
LOYOLA CAMPUS - CONCORDIA UNIVERSITY
7141 SHERBROOKE ST. W.
MONTREAL
(514) 482-0320 x-281

This routine enables FORTRAN to use the .PEEK EMT available
in MACRO. It is usable under RT/TSX. Its function under TSX
is the same as described for the .PEEK EMT in the TSX+
Reference Manual

To use, simply call the FUNCTION from FORTRAN, supplying the proper
arguements in the FUNCTION call
```

```
        EX        IVAL = LMPEEK ( IADDR )

        WHERE     IADDR     I*2 address to PEEK at low memory
                  IVAL      I*2 value PEEKed at in IADDR

        CAUTION             ARGUEMENTS ARE NOT CHECKED. Any wrong paramaters
                            or an non-existant address will cause unpredictable
                            results


LMPEEK: .PEEK     #EMTARR,@2(R5)              ; Take a PEEK - IVAL returned in R0
        RTS       PC                          ; BYE-BYE     - FUNCTION returned R0

EMTARR: .BLKW     2                           ; EMT work area

        .END



        .TITLE    LMPOKE            ;FORTRAN callable .POKE EMT SUBROUTINE
        .IDENT    /V01.00/          ; The first try
        .ENABL    LC

        .GLOBL    LMPOKE
        .MCALL    .POKE

        OCT 18/84
        MIKE MARAK
        DAVID GAUDINE
        EMC LAB
        LOYOLA CAMPUS - CONCORDIA UNIVERSITY
        7141 SHERBROOKE ST. W.
        MONTREAL
        (514) 482-0320 x-281

        This routine enables FORTRAN to use the .POKE EMT available
        in MACRO. It is usable under RT/TSX. Its operation under TSX is
        as described for the .POKE EMT in the TSX+ Reference Manual.

        To use, simply call the SUBROUTINE from FORTRAN, supplying the proper
        arguements in the call

        EX        CALL LMPOKE ( IADDR, IVAL )

        WHERE     IADDR     I*2 address to poke in low memory
                  IVAL      I*2 value to poke into IADDR

        CAUTION             ARGUEMENTS ARE NOT CHECKED. Any wrong paramaters
                            or an non-existant address will cause unpredictable
                            results


LMPOKE: .POKE     #EMTARR,@2(R5),@4(R5)       ; How about a little POKE
        RTS       PC                          ; Slam Bang Thank You Mam

EMTARR: .BLKW     3                           ; EMT work area

        .END
```

DECUS (Australia) 1984 Wish List

David Fingerhut,

Digital Equipment Corporation,

U. S. A.

Dear David,

We enjoyed having you at Decus this year, even if you have sworn never to walk across the Sydney Harbour Bridge again in your life. Herewith as promised the RT-11 Wish List, for your perusal, distribution, and comment. You will note that it has expanded since DECUS, having been posted to most corners of this country for ideas.

BUP:

- The restriction that only one file or device can be backed up onto a magnetic tape severely limits the usefulness of BUP when multiple files (devices) have to be backed up. The most convenient way to perform such a backup would be to mount a blank tape and invoke a command file before going home. Because BUP requires the intervention of an operator to change the tape for each file (device), unattended backup is not possible. The ability to backup multiple files (devices) on one tape would also reduce the cost (for multiple tapes).

- To speed up backup process in situations where there are a couple of "backup" units available (eg floppies), it would be useful to allow specification of a new output device when BUP has filled one up and requires another to continue the backup process. (Conversely on restore, allowing specification of next input device.) This could be implemented by changing the section of the program that prompts for a new output volume to be mounted such that

  MOUNT NEXT VOLUME: (return) continues with previous device, but
  MOUNT NEXT VOLUME: dev:(return) switches to (dev).

  There are several installations here where the floppy drives are in another room from the user's terminal, and to have to wander back and forth 20 times per backup/restore is a little tedious! With this implementation it could all be run from a command file, with the user merely alternating floppies.

- The current scheme for verifying a backup volume is inadequate and allows data to be backed up on faulty media. It would be more secure to backup the data and then perform a binary comparison of the original data and the backup volume.

- A switch to allow the user to wear the consequences of potential bad blocks where he is prepared to trade off reliability against speed would be useful (? /RISKY).

PIP:

- COPY *.* DEV:
  prangs if there is no room for any one file. Generally this is satisfactory, allowing the user to take evasive action, but it would be useful to have a switch allowing an error message to be produced but permitting copying to procede with subsequent files which may fit.

- A single device (or file) name typed as a command to PIP (eg *TT:) causes the weird message "System Error". Unnerving for novices!

- COPY/FROM:filename to select starting point for copying would save many tedious COPY/QUERY "no" answers.

DUP:

- COPY/DEVICE/FILE FILE1.DSK DEV:FILE2.DSK
  has a nasty bug. Instead of producing or overwriting DEV:FILE2.DSK or producing an error message, it successfully overwrites DEV:. Whoops! Our feeling is that it file2.dsk does not exist it should be created, and if it does exist then it should be treated as a device and overwritten up to the lesser of the sizes of the two files. What do you think?

- DUP should check the codes of devices rather than their names to determine which require CLEANing after a squeeze. This would allow more than 8 LD's (by copying LD.SYS to LE.SYS - with relevant patches - etc).

- Ability to shrink files as well as expand them (ie /T with negative argument). This would allow shrinking of logical discs. Note that this facility is supplied with RTEM under the JOAT programme.

MACRO:

- The ability to include ascii text as text in a manner analogous to the ".remark" would tidy up innumerable help texts within programmes (see the Macro-10 .ascii directive). It is a pain

  to have to type a .ASCIZ directive before and append (15)(12) after each of forty or fifty lines of help text.

- To enable the handling of libraries in a civilized fashion, the ability to store a complete library as a single source file containing multiple modules (separated for example with ".prgend" instead of ".end" - again see Macro-10) would be worthwhile. [The .OBJ file would still need LIBR processing.]

- Include the source file line number when error messages (only) are being output to the terminal. The only way of finding an error in a program with a large number of macros is to produce a listing. It would be much easier to have MACRO indicate that the error was on line N. Then it would simply be a matter of editing the file and typing [GOLD][N][0] to position the cursor on the right line (ie the one that has the error).

- For those of us that have to work with 32 bit machines as well as 16 bit machines, would it be possible to support a .RADIX HEX directive. It would be even nicer if a switch or customization patch was supplied so that all output could be produced in HEX instead of OCTAL.

- PUSH and POP supported (either in SYSMAC.SML or as per CALL & RETURN). Would be nice if allow multiple items per instruction (ie .PUSH (R0,R1)).

- Some means of time/date tagging of modules, possibly similar to the ".LIMIT" in concept: eg

  .DATTAG
  could set up a block such as

  .word (date of assembly)
  .word (date of linking)

  the first word being filled by Macro, and the last by Link. [As this would have to work across other DEC operating systems, it may need to be .byte day, month, .word year.]

LINK:

- When an undefined global is detected, please can we have the error message display the module(s) in which it is referenced? Wading through gargantuan link maps is masochistic, especially when one is pushed for time (aren't we all?). Even a switch allowing typing of only the lines from the link cross-reference map containing the undefined globals (/GLOBAL:UNDEFINED) would be a good start.

- A customization or switch to allow the map file to be output in HEX.

- A switch to disable the automatic search of SY:SYSLIB.OBJ. It is ridiculous that the only way of stopping routines from SYSLIB being linked in is to temporarily rename it to something else. (I don't want SYSLIB linked in when I want to check what globals are undefined.)

- Does the .WEAK directive really work as documented? It looked like the answer to a problem we had, but either we misread to documentation or it doesn't work. Could the documentation on .WEAK be improved - it is weak at the moment.

FORTRAN:

- INCLUDE directive, for including additional file in source code (eg common blocks, declarations).

- Type the line in error to the terminal when an error is detected.

TRANSF:

- Would like wildcard support, and magnetic tape commands.

- The TRANSF.SAV distributed should be linked for normal rather than virtual use. [Dissention here: the PC users want it linked virtual!]

- Document /R switch for prompts, and make it possible to access this from DCL (eg TRANSF/HELP).

- Would like a switch to force TRANSF to use decent sized packets (eg 512, 1024). Watching it plodding along on two otherwise unloaded systems with a packet size of eight (and then die at 4am!) was infuriating!

KED:

Of all the RT-11 utilities, KED is the most used and the most important. Because most of us (programmers) spend 50-75% of our time editing, KED has the greatest impact on productivity and "happiness". The big disappointment with RT-11 V5.0 was that there was no significant improvement in KED. In fairness to those of us who pay DEC for full software support, it would be nice to get something for our money. (We use TSX-Plus and so are more interested in the utilities than the monitor.) If DEC isn't going to play fair with KED, can we at least get the sources put into the DECUS library?

- Decent screen handling (a la EDT). In fact, KED & EDT more compatible, retaining any superior features of KED (of course) - see comment on KED LEARN below.

- Ability to search for control characters (and (escape)'s).
- VT200 support (K20?). The VT200 supports 8 bit escape sequences and has a lot more keys. The 8 bit escape sequences can be used to reduce the communications overhead and the extra keys (and better - well, different - key layout) can be used to increase ease of use.

- The user should be able to assign a string to any key on the keyboard. This would allow the keyboard to be optimized for a particular application and would get around KED's limitation of only one LEARN buffer. In effect a "learn" string could be assigned to any unused key (ie "~", "`"). EDT has the ability to redefine key codes! However, the KED LEARN, with its demonstration of what is actually happening at the time, is extremely useful.

- A [GOLD][P] (protect against system crash) function that would close out the file and then reopen it leaving the cursor positioned at the same spot. This process should be as transparent to the user as possible (ie select region should remain etc). To overcome the problem of what the file will be called, the intermediate file should probably be called CRASH.KED, and CRASH.KED should be automatically deleted when KED is exited. EDT has features to protect the user against a system crash.

- A "SET PROTECTION OFF/EVERY N MINUTES/EVERY N CHARS" which performs the same function as [GOLD][P] above automatically

every N minutes or after the user has typed N characters.

- A "SET OVERWRITE" (? [GOLD][o]/[i] - overwrite/insert) command that would allow new input to overwrite existing text rather than being inserted as at present. Overwrite would be most useful when a diagram or flowchart has to be included in documentation, and when working over a slow line.

- A "SET WINDOW START,END" command to allow the size of the display window to be reduced. This would be a godsend for people who have to work over slow telephone lines. Another advantage (if KED didn't clear all the screen on startup) is that we could make use of the VT100 split screen ability to talk to KEX on one part of the screen a background job on another.

- A "SET SCROLL REGION TOP,BOTTOM" so that KED would scroll the screen only if cursor got to within TOP lines to the top or BOTTOM lines to the bottom. When KED has to scroll it could place the cursor in the middle of the screen, minimizing the amount of scrolling done. This would be useful when working over slow lines.

- KED should automatically execute the startup file DK:KED.INI on startup. Why must users manually select the same features each time they edit? (Saving a "tailored" version of KED is no use, as we want different features set up when working with different files: - eg discs [DK:] containing documentation, MACRO or PASCAL files)

- A "SET UNSELECT AUTO/MANUAL" command. In manual mode a "selected" region would remain selected until a [GOLD][.] was pressed. This would allow a region to be selected, filled, converted to lower case and written out to a file without having to be re-selected after each function.

- Make KED configurable for different flavours of terminals (? objects provided, with customizable terminal interface code). Teaching environments in particular cannot afford to buy VT100/VT220's for everyone.

- Backspace to swap previous two characters (similar to SL but different - see below).

- Why does KED do something every 15 seconds or so when it is supposed to be doing nothing? (This matters under TSX, when it causes a swap of the KED job for no apparent reason.) The .MRKT request should be killed whenever waiting for TTY input.

LET/SL:

- Allow spaces in LET strings please.

- Allow control characters in input under SL unless sending line to CSI for parsing (ie allow with GTLIN but not with CSIGEN or CSISPC). It is useful sometimes to include (tab)'s and (esc)'s.

- SET SL KED, or similar ideas to the TSX version [optional, as would enlarge the already large beast - in fact, how about an optional tiny version with no help text for those familiar with it?].

- Backspace swaps the wrong two characters. When a mistake is detected it is usually immediately after typing it, so backspace should swap the character before the pointer, and the character before that. (Here, there would be no need for (GOLD)(backspace) as the action would be the same.)

LD:

- Ability to boot from LD. (Would require offset ability within handler, both in boot section and normal running section.) See below comments about running both RT-11 and POS from a single winchester. [It would not even be necessary to perform a non-LD boot first, but would probably be easier to implement with this restriction.]

- SET option to optionally allow LD to come up with no logical units mounted on reboot, rather than remembering the eight you were using last time (and may have no intention of using this time!).

- Ability to dismount multiple units (eg DISMOUNT LD,LD1,LD4, and/or DISMOUNT *).

- Subroutine or monitor call to allow application programme to mount an LD:, and determine the status of any LD units at any particular time.

DIR:

- Ability to produce nested logical disc directories (assuming a .DSK file will be a disc image file), indenting the files in each subdirectory and sub-sub (etc) directory.

- Wildcard on devices: eg DIR LD*:FRED (checks LD0 - LD7, not reporting on any which cause an error on reading the first directory segment).

- DIR DEV1:,DEV2: You mightn't like it, but we do! Don't even care if it transmutes to

```
.R DIR
*DEV1:
*DEV2:
*^C
```

  changing lines at each change of devices.

- There is a word in the directory used while a file is tentative but unused when a file is made permanent. Change USR so that time of day (in number of 3 second intervals past midnight) is stored in this word when file is made permanent. This feature is provided under TSX-Plus and is extremely useful.

UCL:

- Multiple argument facility: allow ^1, ^2, ^3, ... etc for multiple arguments.

- Support for proper parameter insertion, as in

```
DO:==EDIT ^.FOR
LL:==LIBR MTRX ^/U
```

  which doesn't like the ".FOR" or the "/U".

GENERAL:

- Set TT Hold could be useful on terminals other than VT52's.

- SET DZ NOWRITE was useful - please put it back. It provided a simple way of providing a secure turnkey system. However it would be improved by the ability to protect individual drives

- EG SET DZ0 NOWRITE.

- What is the part number for the binder for the mini-reference manual? (Alternatively, can it be shipped in a binder?)

- Would like the sources for ALL handlers (incl PI, SL) included in normal RT-11 distribution kit. One of the nice things about RT-11 has been its openness to hackers (rather, people with different but well-defined requirements), and customization to PI and SL is most difficult without sources!

- VMS search command (to search for a string across whole device or set of devices) would be useful.

- Would like an <u>Instrumentation Bus Library</u> which works with polled devices:
  Using each of the three available, on 12 devices, currently
  V1.0 8 of 12 work
  V2.0 10 of 12 work
  V2.1 10 of 12 work, but a different 10!

  An IB.SYS which will work unter TSX plus is rather urgently needed.

- <u>Virtual Overlay Handler:</u> if a programme linked for this is used <u>under the FB monitor</u>, the handler should be smart enough to use a disc. In general, programmes which require a specific environment should check for that environment, and if it is not present, either take satisfactory alternative action (preferably), or produce a reasonable error message.

- It is currently difficult to run RT-11 and POS on the same winchester disk. If the RT-11 disk drivers were changed to support a SET BASE B option such that block N was mapped to block N+B, it would be possible to create a contiguous POS file on the winchester and copy the RT-11 system into that logical disk. A simple program could then be provided under POS to boot the special RT-11 file. This would be very useful to PRO users and to PDP-11 users who use more than one operating system (ie RT-11 and RSTS/E). [See also ideas on booting from LD units.]

- It should be able to declare a "INTERRUPT CHARACTER" such that monitor passes control to user specified completion routine if declared character is typed on keyboard.

- Should be able to specify a .TTYIN TIME-OUT delay so that a user specifiable character is returned by the operating system if no input is available from keyboard within time-out period. [Alternatively, a TT: .READC with a word count of zero to tell a programme (via a completion routine) when when input is available without (a) reading anything or (b) having to sit in a tight loop waiting for it.]

- Can the name of the logical area on which the utilities reside be changed from SY: to UT: (or something equivalent). This would allow those with multiple drive floppies to keep all their utilities on line and those with a single big winchester disk to remove a large number of files from the outer-most directory.

- The ability to assign a search path to the logical names SY and DK. A command "SET SEARCH SY DM0,LD0,LD2" would result in a LOOKUP on SY:FILE.XXX searching DM0:, LD0: and LD2: for the file. The search would stop when FILE.XXX was found. This feature would help users with large winchesters. Various types of programs could be grouped into logical disks (ie word processing, accounting, macro programming, C programming) and included within the search path as required. This would allow each logical disk to be kept small and maintainable. (Unfortunately a lot of "packages" require that they run on SY:. Some systems have 1000-2000 files on SY: - this severely slows RT-11 down!)

- VBGEXE: RT11 on the PRO is difficult to use for an experienced user. It is impossible for the inexperienced. All programmes which run under a normal FB monitor on an 11/03 etc should be runnable on a 512kb PRO! - but try the forms editor etc etc.

- Can PIX.SYS run in high memory?

- When did you say RT-32 would be ready for field testing?

Regards,

....	....
Chester Wilson (retiring librarian)	Ray diMarco (librarian ascendant)

From: Carl Lowenstein

TO: RTSIG DISTRIBUTION LIST (@RTSIG)

Subject: TS-11 hangups

We have recently been having problems with various TS-11-type controllers hanging up the system when they sense physical end of tape. Bad news if you're trying to record continuous data and switch to the other tape drive. Anyone else with this experience? Is it RT-11's MS handler, TSX's MS handler, or hardware/firmware glitch?

carl

From: Thomas J. Shinal
To: Carl Lowenstein (LOWENSTEIN)
CC: RTSIG DISTRIBUTION LIST (@RTSIG)

Subject: In Reply To TS-11 hangups
Carl,

We have been having on-going problems with the MS handler under TSX+. The latest release permits us to use the MS Handler in 22 bit mode but we found some restrictions.

Any attempt to abort a tape operation, e.g. abort a DIR or a write operation will hang the system necessitating a reboot.

regards,

Tom

' RT-11 VERSION 10

' Ian Hammond
' HAMMOND software
' An der Lutter 32
' D-3400 G~ottingen
' West Germany

It seems that PDP-11's are going to last a good number of years yet. What with the 11/73 here and optimised CPU's on the way (J11M), any announcement of the death of the PDP-11 would seem premature.

The big VAX's may have the compute power, but they still don't have the numbers. Recent figures from DEC indicate some 20 thousand big VAX's (750, 780). The last figures I remember for PDP-11's were over 300 thousand.

The smaller VAX's have the features of the big guys, but do not offer the performance of the high-end of the PDP-11 range. An 11/73 outperforms an 11/730 in many ways. One of the major problems for the small VAX's is that VMS was developed for a much faster CPU. VMS is largely compute bound on slower processors.

Therefore, I think we can expect a healthy PDP-11 scene. The one thing that will be missing in the future are new large software packages. These will be aimed at the VAX, where fortran programmers can finally get enough space for their wildest compilations.

I have said all of this in order to justify asking what some would judge an academic question: What will version 10 of RT-11 look like? Users have been discussing this_ theme_ at the edges for some time now. A 32-bit version of RT-11. RT-11 V6. An a la MicroPowerPascal version. Single job as a conditional compile of F/B. Change is in the air.

However, if a new architecture is released we will have to live with it until the end of the PDP-11 lifetime. It is important then to look at how that architecture should be able to expand and to grow. Therefore the question: V10?

Some might wonder if it is useful for users to discuss this kind of question. The answer is simple: it is much too late to start discussing it after the event. This should not be seen as a huge wishlist item but rather as a discussion of issues. Future ideas based on the experience of a sometimes painful history.

I am going to present here what I see as some of the central issues that I believe reflect at least some consensus.

' SMALL, FAST & EASY TO USE

This is GORT. Small. Fast. Easy to use:

Put me in a corner. Dabble at my keyboard for a few days. Then forget me. I will work for you forever. You can spend your time reading Datamation.

Do we want V10 to be big, slow and difficult to use?

The word 'small' is crucial. Small is not just beautiful, it predicates speed (unless you are talking about papertape PDP-11 BASIC, which was small and extremely slow. But, then again, the writers had to work without a PDP-11). Architectural complexity tends to grow exponentially (because there are so many related sub-systems). And yet it would seem impossible to imagine a new RT-11 that was simpler, and difficult to forsee one that maintained the current level of complexity.

Small ten years ago had a different than small does today. The typical machine then was about 8k words. 32k words was big. 32k is now small, 128kw normal and 4 megabytes is big (is anyone actually running a 4 megabyte RT-11 yet?). According to these figures, RT-11 could quadruple at least in space.

Is speed still the issue it was? I am personally biased here - I love fast systems. But how do most users feel: do they prefer features to feet?

It seems in retrospect a bad idea to let developers of a new operating system go to work on a machine at the top of the range. The amount of code in VMS becomes visible when you run on a 730 instead of a 780. And yet a new RT-11 would almost certainly be developed on processors of the power of an 11/73. The past makes us hopeful here: a lot of RT-11 development took place on an 11/70.

However, RT-11 has also had a very restricted brief in the past. Except of course in V3. We still have a number of problems left over from V3.

Easy to use. Users of systems like RSX, UNIX & VMS seem to wallow in the mysteries of their systems. I think we should admit it: we could never produce the kind of monster war stories that so often come out of the RSX SIG. We would have solved the problem too quickly to have remembered the pain. Do we want to change this?

I have seen two recent articles where RSX & UNIX users present their systems as easy to use. Hmm. The UNIX writer simply changed the definition: Easy to use is having lots of powerful utilities. That's not ease of use, that's just having lots of powerful utilities. Ease of use is being able to use those powerful utilities easily.

One reason RT-11 is easy to use is because it cuts corners. You can open a file on one channel, delete it on another, and still keep using it on the first. .CLOSE doesn't even check to see if the channel is done with I/O. On the other hand, we don't have to bother about 'locked files' or thousands of minor error messages. We just notice that the directory looks like it is tied in knots (a whodunit & how'_to'_undo'_it for Houdini).

Seriously, one large temptation for a new architecture would be to round these corners out. Programmers are known for liking things neat & tidy (even MACHO-11 programmers).

In rough conclusion: I think GORT has to stay on her or his throne. V10 of RT-11 should not be another name for RSX, VMS or UNIX. The main advantages of RT-11 remain small, fast & easy-to-use.

' PROTECTION

RT-11 is no profalactic system. A flat structure. You want, you get and don't say I didn't tell you so. RT-11 expects all programs to act as if they were part of the system. The error-checking system looks only for blatant mistakes. It makes the same kind of error-checks for programs that it would make for itself.

In the language of other systems, all RT-11 programs are privileged images. Who wants to write an underprivileged image anyway?

All protection ever does is limit system functionality. It does not extend it. Protection is a feature that stops you using other features. The anti-feature.

There is a strong psychological advantage in RT-11's approach. Protection locks you out and it is only natural that you want to fight back. Unprotected systems make you part of the system and the natural response is to cooperate.

Naturally, there is a good argument for having protection on big systems with large numbers of users. But do we want this on RT-11?

' ANY-WHICH-WAY

RT-11 is a flat goto system. None of the niceties of a one-input one-output philosophy. Any-which-way. You can twiddle here, insert a jump there and it all still seems to work.

If RT-11 had rules to obey, we would spend most of time apologising. Apologies usually cost more code than they are worth.

Yet how many of us would like to see the idiosyncrasies of SYSCOM, JSW and the RMON database go away? The question is: could you do without them?

The RMON offsets have a simple and powerful function. It is not simply that they save the code required to manipulate them. The important thing is they do not define how they will be manipulated. Definition is restriction.

The point I am making poorly in this section is that RT-11's corner cutting and undefines produce a system that has a minimal personality. It leaves as many issues open as possible. This means you can define them yourself. And you don't have to go 'around' the system to do so. Once again, you are part of the system.

Should this change? Would it still be RT if it did?

' THINGS I HATE ABOUT RT-11

Did I really write that sub-title?

My idea here was to deal with the obvious items that should change. After ten minutes all I can think of is FILEX & magtape (I forgot BATCH some time ago).

I remembered. What I hate is the USR. As far as I am concerned anything without a USR must be considered new software technology and is acceptable to me as a new architecture for RT-11. I would also like to see KMON stop wriggling about in memory.

But if we are considering a new and different architecture we should ask the question: if there is so little to dislike, why change?

' COMPATIBILITY

Here's a simple one. Should you be able to run RT-11 V5 programs under V10? That is, must a new system supply compatibility with existing RT-11?

Must it supply continuing compatibility or would it be sufficient for V6 & V7 to be compatible? History would seem to show that once a feature is in, it tends to stay in (although MicroVAX just changed that). RT-11 V2 planned desupport for V1 requests.

For some possible architectures, it would seem a bad idea to require support for existing RT-11 in the system. We would not get the best of both worlds.

However, one viable idea would be to support a compatibility with an AME (an applications migration executive). This module would translate existing RT-11 requests, etc. into equivalents in the new system.

This would probably come anyway. RSTS, RSX, VMS and UNIX all support RT-11 environments with this kind of technique. It would seem a shame if new RT did not do the same.

This is the appropriate place to deal with the fact that one reason RT-11 has survived is this portability. RT-11 is the linga franca of PDP-11 systems. We would not want to lose this with a new architecture.

' 32-BIT RT-11

Many users would like to see VAX/RT. It even sounds good. This has an obvious impact on any thoughts for a new 16-bit RT-11.

Let us assume that both were to be implemented. In this case it would probably turn out that the new architectures would contain substantial similarities. This would permit a number of utilities to be developed in parallel for both systems (but hopefully not in BLISS). Command language level compatibilty would mean less work. Finally, the documentation could be combined.

If a VAX/RT were developed first, there would be strong pressure on a new 16-bit architecture to look the same. There would be less pressure in the opposite case. 32-bits is fun and the architects would want to play around with it.

Should a 32-bit system supply an AME for 16-bit RT-11 - existing and new? This would not be difficult (except on machines that did not support compatibility mode). But user wishlists contain thousands of items that are desirable, not difficult and still beyond the limited resources of a development group.

' THE FILE SYSTEM

The file-system is the most hotly debated area of RT-11. RSX people tell me it has no record-locking (neither do most RSX systems - you need RMS for that and RMS is not worth it). You can't extend files or even truncate them. There is no file protection.

But it is ridiculously fast.

I don't know whether you have ever had to run stand-alone VMS BACKUP from TU58's. You don't know what you are missing. Hours of fun watching TU58's spin backward & forward. Even an RSX floppy copy can provide this kind of enjoyment. FILES-11 was designed to wear the rust of the most resiliant disk surface. In fact it's so bad that many RSX & VMS users go through the agony of FLX (which is even worse than FILEX) and use RT format instead. Indeed, VAX's boot from RT-11 format floppies and TU58's.

The RT-11 file-structure has only one major feature missing: the ability to extend a file. The rest, protection, UIC, etc. could easily be added to the existing structure. Not so extension.

But its not always a choice of this or that. I think there is a good case for a new RT-11 to support multiple file-structures. It could support the existing file structure and leave an open interface (like handlers) for others. Other systems call these things ACP's.

In fact, this is almost possible with existing RT-11 with the special directory device feature.

MicroPower PASCAL, RSTS, RSX & VMS all supply some support for the RT-11 file structure. A new RT would have to do at least as well.

' SINGLE-USER

RSTS, RSX, VMS & UNIX (and others) all support multi-user RT-11 capability. Only RT-11 does not (although V5 & V5.1 get fairly close).

On the one hand, my business makes this a delicate subject for me. On the other hand, it puts me in a knowledgeable position to comment on it. I think that the single-user restriction has produced some good things for RT-11. There is always a temptation to go overboard in multi-user systems. It is far easier to justify lack of performance because of multi-user requirements.

More than that, with the exception of V3, RT-11 has always had a fairly restricted brief. I believe this has promoted more discipline with the implementation, if only because there was more time available (I can hear some cruel laughter).

Let us assume there could be a multi-user version. That would not remove the essential requirement for a front-runner single-user system. The single-user system could not be just a poor-boy subset of the multi-user version.

This is a question of attitude. Multi-user should be considered an enhancement of the single-user system rather than single-user being a sub-set of multi-user.

History has mostly good news here. Although most new RT-11 development takes place on the FB & XM monitors, the SJ monitor has remained essentially compatible and has not grown explosively. Indeed, the SJ monitor has benefited from this lack of attention - no fiddling produces no code.

A few years ago a survey showed that less than 5% of RT-11 systems were using XM. SJ was the main system being used (some felt this was because the distribution kit arrived with a bootable SJ and users just didn't change it). But, the fact remains that RT-11 still is DEC's first (and I need not say it) PDP-11 personal computer system. It is still used mostly this way.

' OPERATING SYSTEM FLAVOURS

In the previous sections, I looked at the subject in terms of current RT-11. This was a useful springboard for ideas. But a new RT-11 could look very different.

Digital has its own way of writing operating systems. The two essential features that flavour a DEC style system are the command level interface and completion routines (which are called AST's on the other systems).

A DEC-style process actually executes three different code streams. The DCL interface, the mainline and completion routines. This is built into the VAX hardware and extended into four levels:

kernel	interrupts, most program requests
executive	file system (USR)
supervisor	command language
user	Nobody but you

This is quite different from the UNIX model where a process executes as a single synchronous stream. UNIX systems generally run two processes per user. The shell process inteprets the command language and spawns a sub-process for each program (UNIX calls this a fork).

shell process	command language
forked process	application

MicroPower PASCAL uses a UNIX like model (its a fairly classic model). It supplies a real-time interface by reducing the amount of context a process has so that you can run a lot of them. In this case, each completion routine stream is contained in a process. VAXelan, DEC's newest system, appears to be built on the same kind of principle.

RT-11, RSX & VMS have more similarities here than differences. Indeed, RT-11 is closer to VMS than RSX in some respects. RSX runs some commands as tasks where RT-11 & VMS tend to move within a single DCL environment (its worth noting that RSX started out in life as a run-time system like MicroPower PASCAL). All three systems deal with interrupts & even handlers (or drivers) in much the same way.

The other similarity between VMS and RT-11/SJ/FB (& parts of XM) is that the monitor is in the same address space as the program. This simplifies the work of monitor and makes it faster. It is also the reason we had to wait a couple of versions for a fully functional XM.

The purpose of this very crude, and probably inaccurate, overview is to ask a simple question: should new RT be built in the classic DEC style or go over to the other side?

There is a certain lack of elegance in the traditional DEC style and it creates some major problems of software context. RSX places major limitations on AST's and the VMS executive started kludging in this area fairly early on. There are a number of RT-11 program requests that cannot be issued in completion routines.

But it is very efficient in many other ways. For example, process creation can be a fairly lengthy procedure: you can slow a system down beautifully by having to create one for each program (it's like running IND from floppies).

The DEC-style system is fitted to the shoe it must wear: the typical behaviour of interactive & asynchronous systems. Do we want to change this?

' BACK TO VERSION ONE?

My intention has been to look at the issues: not to suggest a new architecture. But I cannot resist the temptation to state my own preference.

There are two schools of thought:

different drum RT
best of RT

The 'different drum' idea is that new RT should be different, otherwise it won't be new. The 'best of RT' idea is that new RT should be firmly based on existing RT-11, otherwise it won't be RT.

I support the 'best of RT' school. Difference for difference's sake is kind of crazy when you are 95% happy with the existing system and prefer it over available 'different' systems. UNIX is already there.

The idea of the 'best of' school is to go back to version one and re-examine the entire architecture, keeping and enhancing the best and dumping the rest. This would involve a rewrite rather than a patch-up. An AME would support existing applications. Systems that rewrote their software would not need the AME. It could dissappear over time.

Here are some examples of the kinds of decisions:

Dump V1 EMT formats completely. Clean-up the parameter structure of V2 EMT's so that XM doesn't need a new rule for each program request. Change the names to mean something sensible in some cases (like .DEVICE which has nothing to do with devices).

Look at the growth of the program-DCL interface and try reduce the number of different cases. At present we have: exit to command file, special exit to command file, chain, CCL, UCL, command files and STATWD, control files and IND. This could be combined with a new DCL interface to replace CSI.

Restructure the architecture so that KMON was a normal kind of program and sell the USR to somebody.

In fact, RT-11 has already been through something much like this. RT-11 V3 was basically a new system in many ways.

But the result of the 'best of' approach should be a substantially new system. The implementation would be completely new. It would retain a 'virtual' and cultural compatibility with existing RT.

RT-11 VERSION 10

We are in a fairly unique position with RT-11 as an operating system. We're happy. In diminishing markets, corporations tend to like happy customers.

After a number of years of uncertainty about a doubtful future, we have been given the keys to the PDP-11. RSX-11 is a mature product. Who would have believed it. FORTRAN 77 is even available.

There is always the risk that RT-11 could 'attain its planned product goals'. And yet if Digital are going to produce another quarter of a million PDP-11's, this would seem extremely doubtful. Most of them run RT-11.

This brings me back to the original question in a different way. If we assume there will be a V10 is it conceivable that this will still be based on V1? There is really not more than a few bytes left to haggle for the USR. Some of the routines have been patched so often

USER REQUEST

I also have a problem that you may know the answer to. I would like to write FORTRAN programs that access very large arrays, as in ARRAY(10000). With the limited memory that thell/10 has, I cannot do that. Also, since it is not able to accept more memory, I cannot use the usual Virtual facility. The idea I had was to have some sort of disk-based virtual memory on a swapping basis, like VAX VMS. Ie, the system (or subroutine, whatever) would "know" that to get me ARRAY(10000), it has to go back to disk, and pull in a different block of data, and so on. I thought about writing such a piece of code, but it seems such an obvious need that someone must already have done it. I wonder if you can guide me to finding it - perhaps putting it as a request in a MiniTasker.

I really appreciate the functions you have performed for us all, editing the newsletter and so forth. Thanks very much for all of it.

Looking forward to hearing from you.

Sincerely,

Richard Krasnow
c/o J. W. Hastings' lab

Harvard University
The Biological Laboratories
16 Divinity Ave.
Cambridge, MA
02138

I would like to obtain Kermit-11 (DECUS 11-731) on RT-11 format RX01, RX02, or RX50 floppy disks. The current DECUS version comes only on magnetic tape.

Also, now that F77 is available for RT-11, can PORTACALC (DECUS 11-SP-47 or 11-SP-56) be made to work under RT-11? I would be willing to do the conversion if someone can provide the sources on RT-11 floppy disks.

I will supply initialized floppy disks, and any reasonable handling charge to anyone in the SIG who can provide either of these programs.

Sincerely,

Charles M. Moore III
Ecotomic Group, Inc.
P.O. Box 5667
Arlington, TX 76011
(817) 261-0461

DECUS LIBRARY

new
11-761

ODTV09: RT-11 Debugging Tool

Version: V09, October 1983

Author: Donald R. Hanson, Lockheed Aircraft Service, Ontario, CA

Operating System: RT-11 V3, V4, V5

Source Language: MACRO-11

Memory Required: 4.6KW

An expansion of ODT, ODTV09 uses the system terminal or an auxillary to display 35 Application Program symbol values and the GPR's R0 to R5 at program breakpoints. Data is encoded to octal/decimal/hex per user option. It creates a user file of AP symbols and searches the link map for their addresses. Non global symbols may be entered via prompts. Repeat debug sessions input stored file and updates addresses. Symbol file is altered on command. Display format is 3 columns of 12 rows, spaced for clarity. Each column contains the symbol, value, hi and lo byte values. Two additional rows display R0 thru R5.

A dynamic mode allows the AP to run without interuption, updating the display via the LTC. User may SET/CLR symbol locations or transfer interger values via an accumulator "on the fly". A timed blink cycle monitors AP flag (set/clr) conditions. Single keystroke control of 10 global flags for use in debugging. Global breakpoint and relocation registers enable preload at start up for repetitive sessions. Assembles with/without EIS. Display up-date period is approximatly 1 second @9600 baud.

Documentation on magnetic media.

Media (Service Charge Code): RX02 Floppy Diskette (KA),
600' Magtape (MA)

Format: RT-11

new
11-SP-73

Symposium Tape from the Australian RT-11 SIG, Spring 1984, Australia

Version: Spring 1984

Author: Various

Submitted By: Ray Di Marco, SEQEB, Brisbane, Australia

Operating System: RSTS/E, RT-11

Source Language: C, FORTRAN IV, MACRO-11

This tape contains submissions contributed (in general) by members of the Australian RT-11 SIG. The files have been collected into logical sub-disks. The file README.1ST details how files can be extracted by RSTS/E and RT-11 V4 users.

The following is a partial list of file names taken from the collection:

XD, WORDS, BCOP, PIC2, GF, CWC84, SPECS, VIRDSK, PIC3, EI, FILING, VIRTUL, TTLIB, DX, PIC4, FUN, XD, DBSMNG, CVLLIB, 22BIT, LCOM, TSX, XDATCH, RUNOFF, UCLCVL, PIC1, CBITS, TOOLS

No guarantees are made as to the completeness, usability, or quality of the programs on the tape and the material has not been checked or reviewed.

Complete sources not included. Documentation may or may not be included on the magnetic media.

Media (Service Charge Code): 2400' Magtape (PS)

Format: RT-11

new
11-743

DSKLIB: A Disk Librarian Utility Program

Version: V2.18, June 1984

Author: Joel Berez, Berez Associates

Submitted By: James Krugh, Berez Associates

Operating System: RT-11 V2C - V5.1, TSX-PLUS

Source Language: MACRO-11

Memory Required: 16K

Special Hardware Required: EIS is optional.

Disk Librarian (DSKLIB) is a program to maintain a master catalog of multiple disk directories. Disks are given unique numbers and names, and then placed in the master catalog. A powerful FIND command is then used to locate a particular file. Full wildcards and multiple switches are supported. An example of the FIND command is:

```
     FIND file named:  .MAC/DUR MAY/GR 75/LESS 100/PRI
```

This command would print a list of all files in the catalog with the MAC extension created during May of the current year that are greater than 75 blocks, but less than 100 blocks in length.

All RT-11 random access devices are supported.

Documentation on magnetic media.

Media (Service Charge Code): Floppy Diskette (KB),
600' Magtape (MA)

Format: RT-11

new
PRO-126

User Command Linkage-Plus for the PRO-300 Series

Version: V6J, July 1984

Author: William K. Walker, Monsanto Research Corp.,
Miamisburg, OH

Operating System: RT-11 V5.01

Source Language: MACRO-11

Memory Required: 8872 Words

UCL+ is a user command linkage program for use with RT-11 V5.01 on the PRO-300 series computers. It allows dynamic, on-line definition of user commands (or "symbols") and is upward-compatible with the UCL program distributed with RT-11 V5.01. Among the extended features are:

1. An "execute-immediate" mode for commands that are defined in terms of other UCL+ commands.
2. Deletion of multiple symbol definitions in a single command line.
3. Optional chaining to additional "UCL's".
4. A user-definable "run-by-name path" which extends the RT-11 "run-from-SY:" default.
5. Provision to STORE/RECALL program settings to/from a separate".UCL" file.
6. A PASS ON command that allows you to force UCL+ to "pass-on" a given command string to the next "UCL"in the chain (the default mode) or to a program that you specify.
7. DISPLAY of command expansions with or without execution. The DISPLAY command can also be used to output pre-defined ASCII strings to the console, the printer, or some other device/file (handy for sneaky escape sequences).
8. Several useful built-in "hard-wired" commands including a DCL-style RNO command for use with the DECUS RUNOFF (DECUS No. PRO-104) program.
9. Provision for accepting lower-case input (as well as most control characters).
10. Provision to list all program parameters, including symbol definitions; list output may be directed to devices/files other than the console.

The V6J distribution also includes the source text and instructions necessary to create an on-line HELP facility for UCL+.

Documentation on magnetic media.
Media (Service Charge Code): 5 1/4" Floppy Diskette (JA)
Format: RT-11

new
11-768

Canadian Mortgage Calculation Program

Version: July 1984

Author: Soli S. Bamji, National Research Council of Canada,
Ottawa, Ontario, Canada

Operating System: RT-11 V5.1

Source Language: FORTRAN IV

Memory Required: 2KW

The differences between Canadian and American interest terms prevent the application of the American programs to Canadian mortgage calculations. This program calculates the mortgage using the Canadian system.

For the amount of a loan borrowed at a certain rate, it calculates the monthly payment required to repay the loan within the amortization period. It can generate a table to show the amount that is paid as interest and on the principal, the accumulated interest and the balance of the loan remaining after each periodic payment. It also gives the cost of the mortgage in terms of the total interest paid each year and at the end of the amortization period.

This program can also handle the new options that are now available to the Canadian mortgagor, such as the biweekly or weekly payments. It can calculate a new amortization period following a lump-sum payment at the end of each year during the term of the mortgage and the reduced interest cost that would result from such prepayments.

Documentation on magnetic media.

Media (Service Charge Code): Floppy Diskette (KA),
600' Magtape (MA)

Format: RT-11

new
11-759

PSYCHROM: A Calculating Psychrometric Properties Program

Version: V2.0, August 1984

Author: Luther Wilhelm and Robert Freeland, University of Tennessee, Knoxville, TN

Submitted By: Luther Wilhelm

Operating System: RT-11

Source Language: FORTRAN IV

Memory Required: 8KW

PSYCHROM is a package of subroutines for use in calculating psychrometric properties. All values are in SI units. Input combinations of temperature and relative humidity, temperature and dew point, or temperature and wet-bulb temperature may be used with appropriate subroutines to calculate other properties. Properties calculated (including the two used for calculations) are: temperature, wet-bulb temperature, dew-point temperature, relative humidity, enthalpy, specific volume, humidity ratio, degree of saturation, and partial pressure of the water vapor. Computational procedures are based upon recommedations in the ASHRAE Handbook of Fundamentals (1972). A "test" program with data is included to verify and compare routines used for the three possible input combinations.

Documentation on magnetic media.

Media (Service Charge Code): Write-Up and Listing (DA), Floppy Diskette (KA), 600' Magtape (MA)

Format: RT-11

revision
11-490

TSXLIB: A Library Implementation of Programmed Requests for TSX-Plus

Version: 84h09a, August 1984

Author: N.A. Bourgeois, Jr., NAB Software Services, Albuquerque, NM

Operating System: TSX-Plus through V5.Ø

Source Language: MACRO-11

TSXLIB is a library of FORTRAN callable routines that implement the TSX-PLUS system services which are unique to TSX-Plus. The library has been updated to include all TSX-PLUS unique services through TSX-Plus V5.Ø.

Like RT-11, TSX-Plus offers the MACRO programmer a number of system services. These services are implemented via both the RT-11 programmed requests (for those services common to both RT-11 and TSX-Plus) and raw EMT instructions (for those unique to TSX-Plus). RT-11 makes its system services available to the FORTRAN programmer through the system subroutine library, SYSLIB. TSX-Plus also honors the bulk of the service requests in the SYSLIB routines. TSXLIB, however, makes the TSX-Plus unique EMTs available to the FORTRAN programmer.

The TSXLIB distribution kit includes the MACRO-11 source modules for all the routines, a user's manual in machine readable form, and indirect command file to build the library, and the implemented library.

Changes and Improvements: Correction of several bugs.

Documentation on magnetic media.

Media (Service Charge Code): RXØ2 Floppy Diskette (KA),
6ØØ' Magtape (MA)

Format: RT-11

revision
11-564

ADCON: A/D Conversions Package for Use with ADV11-C and KWV11-C

Version: V1.2, August 1984

Author: G.C. Scott

Operating System: RT-11SJ V4.Ø

Source Language: FORTRAN IV, MACRO-11

Memory Required: 6.656KW

Special Hardware Required: ADV11-C analog-to digital conversion pc board, KWV11-C programmable realtime clock pc board.

ADCON is a software package which can be used with Digital Equipment Corporation's ADV11-C analog-to digital conversion and KWV11-C programmable realtime clock boards. The characteristics of this software package include:

(1) digitizations from 8 differential analog input channels,
(2) up to 1 KHz sampling rate,
(3) software programmable gain,
(4) 12 bit data resolution, and
(5) DC offset corrections and calibrations of the data.

The package includes the following:

(1) DY1:DOCA.TXT, user instructions and programming example,
(2) DY1:ADCONF.FOR, a FORTRAN IV program which sets up for the A/D conversions and performs corrections and calibrations of the digitized data,
(3) DY1:ADCONM.MAC, a MACRO-11 program which controls the A/D conversions, and
(4) listing files for the above FORTRAN IV and MACRO-11 programs.

Error checking is performed throughout DY:ADCONF.FOR and DY:ADCONM.MAC. Suggestions for software and hardware setup are included for users who require more customized A/D conversion schemes.

Changes and/or Improvements: Current version speeds up conversion process considerably and eliminates need of DY1:CHANGE.FOR and DUMP utility.

Documentation on magnetic media.
Media (Service Charge Code): Floppy Diskette (KA),
600' Magtape (MA)
Format: RT-11

new
11-757

DACON: D/A Conversions Package for Use with AAV11-C and KWV11-C

Version: V1.0, August 1984

Author: G.C. Scott

Operating System: RT-11SJ V4.0

Source Language: FORTRAN IV, MACRO-11

Memory Required: 4.928KW

Special Hardware Required: AAV11-C digital-to-analog-conversion pc board, KWV11-C programmable realtime clock pc board.

DACON is a software package which can be used with Digital Equipment Corporation's AAV11-C digital-to-analog conversion and KWV11-C programmable realtime clock boards. The characteristics of this software package include:

(1) D/A conversion from 4 differential digital input channels,
(2) up to 1 KHz sampling rate,
(3) 12 bit data resolution, and
(4) dc offset corrections and calibrations of the data.

The package includes the following:

(1) DY1:DOCD.TXT, user instructions and programming example,
(2) DY1:DACONF.FOR, a FORTRAN IV program which sets up for the D/A conversions and performs corrections and calibrations of the input digital data,
(3) DY1:DACONM.MAC, a MACRO-11 program which controls the D/A conversions, and
(4) listing files for the above FORTRAN IV and MACRO-11 programs.

Error checking is performed throughout DY1:DACONF.FOR and DY1:DACONM.MAC. Suggestions for software and hardware setup are included for users who require more customized A/D conversion schemes.

Documentation on magnetic media.

Media (Service Charge Code): Floppy Diskette (KA),
600' Magtape (MA)

Format: RT-11

new
11-758

REVISE: A Scanner for RT-11 Device Directories

Version: V1.0, September 1984

Author: Dr. Peter A. Stockwell, University of Otago, Dunedin,
New Zealand

Operating System: RT-11

Source Language: PASCAL

Memory Required: 28KW or less

REVISE scans through RT-11 device directories for files matching given wildcard specifications and gives the user the option of inspecting or deleting each file in turn. It provides a straight-forward method for examining a whole series of files and deleting those which are no longer of interest. This function is particularly useful for clearing up an accumulation of out-dated or garbage files on larger RT-11 volumes.

REVISE is largely based in concept on the REV program for DECsystem 10/20 (DECUS No. 10-289) but lacks some of the more sophisticated features of the latter (i.e. no rename or back functions).

The sources for REVISE contain a wild card string matching Pascal function and contain Pascal definitions of RT-11 directory structure with routines for opening a given directory and scanning successive entries. They also contain a Pascal routine for converting RT-11 directory date entries into DD-Mmm-YY format.

Restrictions: Sources make use of features specific to Oregon Software Pascal-2 V2.1 or later.

Documentation on magnetic media.

Media (Service Charge Code): Write-up (AA), Floppy Diskette (KA),
600' Magtape (MA)

Format: RT-11

revision
11-736

Virtual DECtape System for Version 4 of RT-11

Version: October 1984

Author: R. Tapp, UVIC Academic Systems, Victoria, B.C., Canada

Operating System: RT-11 V4, VAX/VMS V3.6

Source Language: MACRO-11, PASCAL

Memory Required: 32KB

Special Hardware Required: DL-11 or compatible serial interface

A serial interface to host computer's terminal port provides file storage for an RT-11 system. Included are a virtual-terminal program for RT-11 that runs as a foreground job, a source patch to create the virtual-DECtape handler from the TU58-DECtape handler source file, DD.MAC, and a companion virtual-DECtape file server program for a VAX/VMS host with source code in PASCAL. RT-11 files are stored in a DEC-tape image file on the host computer's disk, which the RT-11 system accesses by means of the virtual-DECtape handler communicating with the file server program that runs on the host. The file server program can also transfer VMS files into or out of the DECtape images, thereby providing file communication between the VMS and RT-11 systems.

Note: Operating system version dependent because device handler is written to RT-11 version 4 specifications.

Documentation on magnetic media.

Media (Service Charge Code): Floppy Diskette (KA),
6ØØ' Magtape (MA)

Format: RT-11

new
11-765

CALEND: A Calendar Program

Version: V1.Ø, October 1984

Author: James H. Norman, White Sands Missle Range, NM

Operating System: RSX-11M V3.2, RT-11 V4.Ø, VAX/VMS V3.4

Source Language: FORTRAN IV

Memory Required: 6144KB

A calendar is produced for any year between 19Ø1 and 2Ø99. The user is prompted for the year. The calendar is then written to a file named CALEND.PRT after which the program exits. The print file may be written to a line printer or a terminal.

Documentation on magnetic media.

Media (Service Charge Code): Floppy Diskette (KA),
6ØØ' Magtape (MA)

revision
11-49Ø

TSXLIB: A Library Implementation of EMTs for TSX-PLUS V5.1

Version: 84j18a, October 1984

Author: N. A. Bourgeois, Jr., NAB Software Services, Inc.,
Albuquerque, NM

Operating System: TSX-PLUS V5.1

Source Language: MACRO-11

Special Hardware Required: MMU to support TSX-PLUS

TSXLIB is a library of FORTRAN callable routines that implement the TSX-PLUS system services which are unique to TSX-PLUS. The library has been updated to include all TSX-PLUS unique services through TSX-PLUS V5.1.

Like RT-11, TSX-PLUS offers the MACRO-11 programmer a number of system services. These services are implemented via both the RT-11 programmed requests (for those services common to both RT-11 and TSX-PLUS) and raw EMT instructions (for those unique to TSX-PLUS). RT-11 makes its system services available to the FORTRAN programmer through the system subroutine library, SYSLIB. TSX-PLUS also honors the bulk of the service requests in the SYSLIB routines. TSXLIB, however, makes the TSX-PLUS unique EMTs available to the FORTRAN programmer.

These TSX-PLUS library routines provide facilities to support communication lines, detached jobs, device allocating and deallocating, file structured device mounting and dismounting, communication between running programs, job status monitoring, program performance analysis, realtime program execution, shared run time systems, shared files, special files information, spooler control, communication between running programs and a terminal, ODT activation mode, user name control, and several miscellaneous EMTs.

The TSXLIB distribution kit includes the MACRO-11 source modules for all the routines, a user's manual in machine readable form, an indirect command file to build the library, and the implemented library.

Changes and/or Improvements: Updated for TSX-Plus V5.1 and some bug fixes.

Documentation on magnetic media.

Media (Service Charge Code): RXØ2 Floppy Diskette (KA),
6ØØ' Magtape (MA)

Format: RT-11

PAST SYMPOSIUM INFORMATION

EDUCATIONAL CASSETTE PROGRAM

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY

Valuable information
Recorded "Live"

DECUS

December 10-14, 1984
Anaheim Convention Center

1984 FALL DECUS U.S. SYMPOSIUM

$7.50 PER CASSETTE

(some sessions presented on 2 or more cassettes)

Account Number
84267-

APL SIG

___ A001F84 WHY USE APL? ($7.50)

BASIC SIG

___ B005F84 A GENERALIZED BASIC ERROR PROCESSOR ($7.50)
___ B006F84 COMMUNICATING BETWEEN 2 PC & BETWEEN APC & APDP 11/70 ($7.50)
___ B007F84 FUTURE DIRECTION OF BASIC-11/RT-11 ($7.50)
___ B008F84 VAX BASIC & MICROVMS ($7.50)
___ B009F84 MODULAR PROGRAMMING IN VAX BASIC ($7.50)
___ B010F84 BASIC-PLUS-2 RUN TIME OPTIMIZATION ($7.50)
___ B011F84 NEXT RELEASE OF BASIC-PLUS-2 ($7.50)
___ B012F84 MEMORY RESIDENT OVERLAYS IN BASIC-PLUS-2 PROGRAMS ($7.50)
___ B014F84 CONVERTING BASIC FIELD STATEMENTS TO REMAP ($7.50)

COBOL SIG

___ C012F84 COBOL APPLICATION QUESTIONS & ANSWERS ($7.50)
___ C013F84 BUSINESS APPLICATION ($15.00)
___ C014F84 WHY USE THE BUSINESS LANGUAGES? ($7.50)
___ C015F84 SCREEN HANDLING FROM VAX RPG 11 ($7.50)
___ C016F84 DEBUGGING VAX RPG 11 PROGRAMS ($7.50)
___ C017F84 THE VAX RPG 11 EDITOR ($7.50)
___ C018F84 VAX RPG 11 MIGRATION/COEXISTENCE ($7.50)
___ C019F84 CONVERTING PDP-11 COBOL PROGRAMS ($7.50)
___ C020F84 USING SYSTEM SERVICES FROM COBOL-81 PROGRAMS ($7.50)
___ C021F84 CONVERTING COBOL-11 TO COBOL-81 ($7.50)

DATA MANAGEMENT SYSTEMS SIG

___ DM009F84 ADVANCED FMS PROGRAMMING TECHNIQUES
___ DM014F84 DBMS-20 V6.1 ($7.50)
___ DM015F84 CRITERIA FOR SELECTING YOUR RELATIONAL DATABASE ($7.50)
___ DM016F84 SCIENTIFIC DATA BASES ON A VAX 11/780 RUNNING VMS ($7.50)
___ DM019F84 CONVERTING EXTERNAL DATA TO A VTX DATABASE ($7.50)
___ DM020F84 DESIGNING EXTERNAL APPLICATIONS FOR VTX ($7.50)
___ DM021F84 CONSTRUCTING A CORPORATE VIDEOTEX DATABASE ($7.50)
___ DM022F84 VTX OVERVIEW AND UPDATE ($7.50)
___ DM024F84 OPTIMIZATION OF VAX-11 RMS INDEXED FILE PERFORMANCE ($7.50)
___ DM025F84 PERFORMANCE OPTIMIZATION UNDER RDB/VMS RELATIONAL DATABASE ($7.50)
___ DM026F84 BUILDING LARGE SYSTEMS USING VAX-11 CDD, TDMS AND COBOL ($7.50)
___ DM027F84 A VAX-11 DATATRIEVE/TDMS/FORTRAN APPLICATION ($7.50)
___ DM029 ADVANCED PROGRAMMING WITH VAX FMS ($7.50)
___ DM030F84 EXPERIENCES OF A VAX INFORMATION ARCHITECT ($7.50)
___ DM032F84 TDMS/DTR IN VAX-11 DBMS APPLICATIONS ($7.50)
___ DM033F84 IMPLEMENTING MULTI-FORM SCREENS USING FMS ($7.50)
___ DM035F84 DATA MANAGEMENT FOR A NURSE'S STATION DATA BASE ($7.50)
___ DM040F84 WHY USE VAX CDD? ($7.50)
___ DM053F84 HOW TO SELECT A DBMS ($7.50)
___ DM056F84 VAX RDB/VMS AND BRITTON-LEE IDM 500 PERFORMANCE ANALYSIS ($7.50)
___ DM057F84 PROGRAMMING EFFICIENTLY IN VAX LISP ($7.50)
___ DM058F84 PROGRAMMING IN OPS5 FOR VAX ($7.50)
___ DM059F84 GETTING STARTED IN ARTIFICIAL INTELLIGENCE ($7.50)
___ DM060F84 OVERVIEW OF SOME A1 ADVANCED DEVELOPMENT AT DEC ($7.50)
___ DM065F84 RELATIONAL DOCUMENT MANAGEMENT FOR

___ DM068F84 PANEL: DEC'S RELATIONAL DATABASE MANAGEMENT ($7.50)
___ DM069F84 A NON-TECHNICAL USER INTERFACE TO RDB ($7.50)
___ DM070F84 ELECTRONIC MEETINGS FOR ORGANIZATIONAL EFFECTIVENESS THROUGH COMPUTER TELECONFERENCING ($7.50)
___ DM071F84 INTRODUCTION TO THE VAX RDB PRODUCT SET ($7.50)
___ DM072F84 THE VAX INFORMATION ARCHITECTURE: AN OVERVIEW ($7.50)
___ DM074F84 VAX CDD TECHNICAL TUTORIAL ($7.50)
___ DM075F84 INTRODUCTION TO VAX ACMS APPLICATIONS ($7.50)
___ DM076F84 VAX ACMS INTERNALS ($7.50)
___ DM077F84 VAX ACMS-PERFORMANCE AND APPLICATION DESIGN TRADE-OFFS ($7.50)
___ DM078F84 VAX TDMS OVERVIEW ($7.50)
___ DM079F84 VAX TDMS TECHNICAL TUTORIAL & PERFORMANCE CONSIDERATIONS ($7.50)
___ DM080F84 TO DB OR NOT TO DB: TRADE-OFFS OF FILE MANAGEMENT DATA BASE MANAGEMENT ($7.50)
___ DM081F84 VAX DBMS-LOADING AND UNLOADING DATABASES ($7.50)
___ DM082F84 VAX DBMS-TUNING FOR PERFORMANCE GAINS ($7.50)
___ DM083F84 VAX DBMS V2.0 INTERNALS ($7.50)
___ DM085F84 RDB/VMS TECHNICAL TUTORIAL ($7.50)
___ DM097F84 AN ARTIFICIAL INTELLIGENCE PROJECT CASE STUDY — THE FIRST SIX MONTHS ($7.50)
___ DM099F84 GENERIC FMS SCREEN PROCESSOR ($7.50)
___ DM100F84 MIS WORKSHOP III ($15.00)

DATATRIEVE SIG

___ DT005F84 EXPLOITING DATATRIEVE TO ENHANCE DEVELOPMENT OF A MICROELECTRONICS RESEARCH COMMUNITY IN THE STATE OF NORTH CAROLINA ($7.50)
___ DT006F84 DATATRIEVE RECORD DEFINITION WORKSHOP ($7.50)
___ DT007F84 SO WHY USE DATATRIEVE ANYWAY? ($7.50)
___ DT008F84 COMPARISON OF DATATRIEVE TO OTHER DATA MANAGEMENT LANGUAGES ($7.50)
___ DT009F84 ON THE EDUCATION OF NEW DATATRIEVE USERS ($7.50)
___ DT011F84 HOW SYSTEM MANAGERS USE DATATRIEVE ($7.50)
___ DT012F84 USING DATATRIEVE AND FMS FOR SOPHISTICATED ONLINE DATA ENTRY ($7.50)
___ DT014F84 VAX-11 DATATRIEVE TO RAINBOW 100 COMMUNICATIONS ($7.50)
___ DT015F84 DATATRIEVE APPLICATION DESIGN WORKSHOP ($7.50)
___ DT016F84 A DATATRIEVE APPROACH TO LABORATORY TEST DATA MANAGEMENT AND PROCESSING ($7.50)
___ DT017F84 CAN YOU REALLY DO THAT IN DATATRIEVE? ($7.50)
___ DT018F84 DATATRIEVE-20 ($7.50)
___ DT019F84 VMS ACCOUNTING USING VAX-11 DATATRIEVE ($7.50)
___ DT020F84 DATATRIEVE-11 FAMILY SET ($7.50)
___ DT021F84 DATATRIEVE AND THE POLICE ($7.50)
___ DT022F84 DATATRIEVE AND A PERSONNEL DATABASE-A CASE STUDY ($7.50)
___ DT023F84 DESIGNING USER FRIENDLY SYSTEMS ($7.50)
___ DT026F84 INTEGRATING DATATRIEVE WITH THE VAX INFORMATION ARCHITECTURE ($7.50)
___ DT027F84 SO, WHAT IS DATATRIEVE ANYWAY? ($7.50)
___ DT028F84 CUSTOMIZING VAX-11 DATATRIEVE ($7.50)
___ DT029F84 USING VAX-11 DATATRIEVE GRAPHICS ($7.50)
___ DT030F84 DATATRIEVE DISTRIBUTED MANIPULATION FACILITY ($7.50)
___ DT031F84 VAX-11 DATATRIEVE-APPLICATION DESIGN TRADE-OFFS ($7.50)

DIBOL BUSINESS SIG

___ DI003F84 LEGAL PROBLEMS RELATED TO COMPUTER SOFTWARE AND HARDWARE WARRANTIES AND TRADE SECRETS VS. COPYRIGHT/PATENT RIGHTS ($7.50)
___ DI006F84 MIGRATING APPLICATIONS TO A TO Z — A CASE HISTORY ($7.50)
___ DI007F84 ACCOUNTING SOLUTIONS ON DIGITAL COMPUTERS ($7.50)
___ DI008F84 DIGITAL ACCOUNTING SYSTEMS (DAS) ($7.50)
___ DI009F84 DECTAP, DIGITAL TAILORED APPLICATION PACKAGES ($7.50)
___ DI010F84 DECTAP ARCHITECTURE ($7.50)
___ DI012F84 MIGRATING TO VAX DIBOL ($7.50)
___ DI013F84 HANDWRITTEN DATA ENTRY ($7.50)
___ DI014F84 NON-DEC DIBOL IMPLEMENTATIONS-USER PANEL ($7.50)
___ DI016F84 PACKAGING TURNKEY SYSTEMS ($7.50)
___ DI017F84 THE BEST OF DIBOL-A WORKSHOP ($7.50)
___ DI018F84 DIBO COS-310 PROGRAMMING PROBLEMS AND TOPICS AND VERSION 9.3 USER FEEDBACK ($7.50)
___ DI019F84 OS/278 RELEASE TO DECUS LIBRARY-USER FEEDBACK PANEL DISCUSSION ($7.50)
___ DI020F84 DECFORM FORUM ($7.50)
___ DI021F84 DIBS-11 TIPS FOR A SUCCESSFUL INSTALLATION ($7.50)
___ DI023F84 INTRODUCTION TO A TO Z ($7.50)
___ DI024F84 A TO Z PRODUCT PANEL ($7.50)
___ DI025F84 A TO Z PROGRAM CERTIFICATION ($7.50)
___ DI026F84 A TO Z LAYERED PRODUCTS PROGRAMMER'S INTERFACE ($7.50)
___ DI027F84 DIBOL AND A TO Z MENU FACILITY ($7.50)
___ DI028F84 DIBOL-83 STATE OF THE LANGUAGE ($7.50)
___ DI029F84 BEYOND DIBOL-83 ($7.50)
___ DI030F84 DIBOL FEEDBACK ($7.50)
___ DI033F84 DIGITAL'S STRATEGY FOR BUSINESS COMPUTING ($7.50)
___ DI034F84 NETWORKING FOR NOVICES USING A TO Z ($7.50)
___ DI036F84 AN ARCHITECTURE FOR USER INFORMATION: CONCEPTS AND PHILOSOPHY ($7.50)
___ DI037F84 IMPLEMENTING AN ON-LINE USER INFORMATION UTILITY ($7.50)
___ DI040F84 TSX-Plus MAGIC ($7.50)

EDUSIG SIG

___ E005F84 IMPLEMENTING AN OPTICAL MARK SENSE GRADE REPORTING SYSTEM ($7.50)
___ E006F84 LAYOUT: A GRAPHICS LANGUAGE DESIGNED BY ARTISTS FOR ARTISTS ($7.50)
___ E007F84 REQUIEM FOR A FRIENDLY EDITOR ($7.50)
___ E008F84 TEACHING AP COMPUTER SCIENCE USING A DEC MINI COMPUTER ($7.50)
___ E009F84 A UNIFIED, MENU-DRIVÉ HELP SYSTEM FOR TOPS-20 ($7.50)
___ E010F84 AN INSTRUCTIONAL/ADMINISTRATIVE RAINBOW SYSTEM FOR A LIBERAL ARTS COLLEGE ($7.50)
___ E011F84 INSTRUCTIONAL COMPUTER GRAPHICS FOR PROBABILITY STATISTICS ($7.50)
___ E014F84 ADMINSTRATIVE SOFTWARE FOR EDUCATION PART 2 ($7.50)
___ E016F84 VAXING AT A SMALL COLLEGE ($7.50)
___ E017F84 ACADEMIC VAX MANAGEMENT PANEL ($7.50)
___ E018F84 ADA: TEACHING THE TEACHERS ($7.50)
___ E019F84 CAMPUS MADNESS-99 WAYS TO DEAL WITH THE MICRO EXPLOSION ($7.50)
___ E020F84 FACULTY AND COMPUTERS — GETTING GOING AND SURVIVING ($7.50)

___ E021F84 SYSTEM MANAGEMENT AT A LARGE VAX EDUCATIONAL SITE ($7.50)

___ E023F84 THE POTENTIAL FOR APPLYING CBE TO REMEDIAL EDUCATION ($7.50)

GRAPHIC APPLICATIONS SIG

___ G001F84 A VAX-BASED GRAPHICS MAIL SYSTEM ($7.50)

___ G002F84 INTERACTIVE GRAPHICS SCENARIO GENERATION ($7.50)

___ G003F84 GRAPHICS FROM THE PERSONAL PRINTER TO LASER TECHNOLOGY ($7.50)

___ G004F84 LASER PRINTERS & TYPE DESIGN ($7.50)

___ G006F84 VAXSTATION DISPLAY SERVICES ($7.50)

___ G007F84 AN EFFICIENT INTERACTIVE MENU HANDLER FOR VT-100/200 DISPLAYS ($7.50)

___ G011F84 ANSI GRAPHICS STANDARDS UPDATE ($7.50)

___ G012F84 GKS AND CORE-SIMILARITIES AND DIFFERENCES ($7.50)

___ G013F84 INTERACTIVE GRAPHICS ON VAX ($7.50)

___ G014F84 REAL TIME APPLICATION OF VAX COLOR GRAPHICS ($7.50)

___ G015F84 VAX GKS/OB TECHNICAL OVERVIEW ($7.50)

HARDWARE MICRO SIG

___ H001F84 QBUS QUESTIONS AND ANSWERS ($7.50)

___ H003F84 Q-BUS/UNIBUS FOREIGN PERIPHERALS FORUM ($7.50)

___ H007F84 HARDCOPY I/O PRODUCT UPDATE ($7.50)

___ H009F84 AN INTELLIGENT COMMUNICATIONS PROCESSOR FOR VAX AND PDP-11: A MICRO/T, MICRO/11 DESIGN APPLICATION ($7.50)

___ H010F84 DEC BUS ARCHITECTURE FOR ASPIRING SYSTEM PROGRAMMERS ($7.50)

___ H013F84 COMMUNICATIONS INTERFACING ($7.50)

___ H014F84 Q-BUS/UNIBUS HARDWARE HINTS & KINKS ($15.00)

___ H015F84 PDP-11 DIRECTIONS AND NEW PRODUCT REVIEW ($7.50)

___ H017F84 CONFIGURING RULES FOR MICRO/PDP-11 SYSTEMS AND OPTIONS ($7.50)

___ H018F84 WHAT DIGITAL WANTS TO KNOW ABOUT MICRO SYSTEMS AND UNIBUS-11'S BUT WAS AFRAID TO ASK ($7.50)

___ H019F84 SHORT TOPICS IN PDP-11 ARCHITECTURE ($7.50)

___ H020F84 HARDWARE SUPPORT FOR MULTIPROCESSING PDP-11'S ($7.50)

___ H021F84 PDP-11 SYSTEMS HARDWARE PANEL ($7.50)

___ H026F84 A REAL-TIME PERSONAL INDENTITY VERIFICATION SYSTEM ($7.50)

___ H027F84 EXCEPTION HANDLING IN MICROPOWER/PASCAL ($7.50)

___ H028F84 MICROPOWER/PASCAL FUTURES AND PRODUCT PANEL ($7.50)

___ H029F84 PROGRAMMING THE KXT-11CA WITH MICROPOWER/PASCAL ($7.50)

___ H030F84 MIL-SPEC AND RUGGEDIZED SYSTEMS ($7.50)

___ H031F84 MULTITASKING WITH MICROPOWER/PASCAL ($7.50)

___ H032F84 MICROPOWER/PASCAL PROGRAM BUILD PROCESS ($7.50)

___ H033F84 INTERFACING I/O DEVICES WITH MICROPOWER/PASCAL ($7.50)

___ H035F84 A PROM-BASED ENERGY MANAGEMENT SYSTEM USING THE FALCON AS A STAND ALONE CONTROLLER ($7.50)

___ H038F84 DESIGN OF AN SBC USING THE DCJ11 MICROPROCESSOR ($7.50)

___ H039F84 MICRO/PDP-11 AND MICRO/VAX UPGRADE OPTIONS ($7.50)

___ H040F84 Q-BUS MASS STORAGE ISSUES AND CONSIDERATIONS ($7.50)

___ H043F84 UPGRADING YOUR 18 BIT LSI-11 SYSTEM TO 22 BIT ($7.50)

___ H044F84 KXT-11 C 10P APPLICATION SHORTS ($7.50)

___ H046F84 LSI-11 PRODUCT INFORMATION FORUM ($7.50)

___ H047F84 LOW COST LSI-11 UPGRADES DEC DOESN'T TELL YOU ABOUT ($7.50)

___ H049F84 ELECTRONIC MEMORY OVERVIEW ($7.50)

___ H051F84 LOW END STORAGE PRODUCT ADVANCES ($7.50)

___ H052F84 CURRENT TAPE PRODUCTS ($7.50)

___ H053F84 OPTICAL DISK TECHNOLOGY ($7.50)

___ H056F84 VIDEO TERMINALS FEEDBACK AND NEW FEATURES ($7.50)

___ H057F84 XXDP V2.0 TUTORIAL ($7.50)

IAS SIG

___ I006F84 AN/GYQ-21 (V) USERS' FORUM ($7.50)

___ I008F84 A STANDARD TASK DUMP SERVICE FOR IAS ($7.50)

___ I010F84 SRX, A NEW DIRECTORY MANAGEMENT TOOL ($7.50)

___ I012F84 HANDLER/LIM EXECUTIVE: A DEVICE HANDLER DEVELOPMENT TOOL ($7.50)

___ I014F84 USING SUPERVISOR MODE LIBRARIES ($7.50)

___ I015F84 IAS PRODUCT PANEL ($7.50)

___ I016F84 UNDERSTANDING IAS CRASH DUMPS ($7.50)

___ I020F84 MACRO-11 SUPER MAC ($7.50)

___ I021F84 LOAD MANAGEMENT-AN IAS F-77 SYSTEM ($7.50)

___ I022F84 BLOCK MODE TERMINALS WITH IAS ($7.50)

LABS/DAARC SIG

___ LA002F84 A SAMPLE TRACKING SYSTEM WITHIN A CHEMICAL R&D ENVIRONMENT ($7.50)

___ LA003F84 REALTIME FAST DATA ACQUISITION USING AN MBD AND CAMAC ($7.50)

___ LA006F84 DIGITAL FILTERS ($7.50)

___ LA007F84 A RESEARCH SYSTEM FOR EVOKED POTENTIAL ANALYSIS ($7.50)

___ LA008F84 RS/1 RMS AND RS/1 DATATRIEVE INTERFACES ($7.50)

___ LA010F84 TS/1 USER PANEL ($7.50)

___ LA011F84 RS/1 AND ALL-IN-ONE ($7.50)

___ LA012F84 RS/1 ORACLE INTERFACE ($7.50)

___ LA013F84 SPEAKEASY ON THE VAX ($7.50)

___ LA014F84 A USER-FRIENDLY DATA ACQUISITION AND ANALYSIS PACKAGE ($7.50)

___ LA015F84 COMPUTER INTERFACING, HOOKING UP TO THE WORLD ($7.50)

___ LA016F84 LIMS GRAPHICS USING REGIS AND DBMS ($7.50)

___ LA017F84 A DATABASE SYSTEM FOR DEVELOPING PROCESS CONTROL SYSTEMS ($7.50)

___ LA018F84 DIGITAL'S STRUCTURED REAL-TIME DEVELOPMENT TOOLS: MICROPOWER/PASCAL AND VAX ELN ($7.50)

___ LA019F84 WHAT IS LIMS/SM, AND HOW DOES IT FIT INTO OVERALL LABORATORY INTEGRATION? ($7.50)

___ LA020F84 VAX LIMS SAMPLE AND INSTRUMENT MANAGEMENT ($7.50)

___ LA021F84 THE ARCHITECTURE OF VAX LABORATORY INFORMATION MANAGEMENT SYSTEM/SAMPLE MANAGEMENT ($7.50)

___ LA022F84 HOW TO CUSTOMIZE YOUR LIMS/SM SYSTEM ($7.50)
___ LA023F84 INTRODUCTION TO REAL-TIME DATA ACQUISITION AND PROCESSING ($7.50)
___ LA024F84 CONFIGURING A SYSTEM FOR REAL-TIME DTA ACQUISITION ($7.50)
___ LA026F84 REAL-TIME DATA ACQUISITION WITH THE PROFESSIONAL PERSONAL COMPUTER ($7.50)
___ LA028F84 REAL-TIME DATA ACQUISITION ON THE PC350 ($7.50)
___ LA029F84 COMPUTATIONAL CHEMISTRY ($7.50)
___ LA030F84 VAX LIMS/DATA MANAGEMENT ($7.50)

LARGE SYSTEMS SIG

___ LS004F84 TOPS 10 Q&A ($15.00)
___ LS005F84 TOPS 20 Q&A ($15.00)
___ LS007F84 DECSYSTEM-10 & DECSYSTEM-20 MANAGEMENT WORKSHOP ($7.50)
___ LS009F84 LARGE SYSTMS TOWN MEETING ($7.50)
___ LS010F84 TOPS-20 VERSION USER PANEL ($7.50)
___ LS011F84 ANF-10 (DEC-10) NETWORK PERFORMANCE ANALYSIS ($7.50)
___ LS013F84 TOPS-10/20 LAYERED PRODUCTS UPDATE ($7.50)
___ LS014F84 FORTRAN-10/20 TECHNICAL SESSION ($7.50)
___ LS015F84 LARGE SYSTEMS PRODUCT PANEL ($7.50)
___ LS016F84 SNA-20 PSI-20 UPDATE ($7.50)
___ LS018F84 TOPS-10 MONITOR DIRECTIONS ($7.50)
___ LS019F84 TOPS-10 CRASH ANALYSIS ($7.50)
___ LS020F84 TOPS-10 UTILITY CLOSET ($7.50)
___ LS023F84 TOPS-20 UTILITY CLOSET ($7.50)
___ LS025F84 VMS V4 FEATURES FOR TOPS USERS ($7.50)
___ LS026F84 TOPS-10/20 TO VAX/VMS PERFORMANCE COMPARISON ($7.50)
___ LS029F84 SNA-20 AND PSI 10/20 UPDATE ($7.50)
___ LS030F84 TOPS-10 7.02/DECNET-10 USERS PANEL ($7.50)
___ LS031F84 TOPS-10/TOPS-20 AND INTEGRATION DOCUMENTATION STATUS ($7.50)
___ LS032F84 TOPS-20 MONITOR DIRECTIONS ($7.50)
___ LS033F84 TOPS-20 NETWORK DIRECTIONS ($7.50)
___ LS034F84 HISTORY OF TOPS-10/20 (OR LIFE IN THE FACT ACS) ($7.50)
___ LS038F84 DECNET APPLICATION INTEGRATION ($7.50)
___ LS041F84 DECSYSTEM 2065 PERFORMANCE PANEL ($7.50)
___ LS042F84 LASER PRINTERS FOR DECSYSTEM 10/20'S ($7.50)
___ LS047F84 36-BIT MAGIC AND WAR STORIES ($7.50)
___ LS048F84 36-BIT TRIVIA BOWL ($7.50)
___ LS049F84 36-BIT PIONEERS ROUND TABLE ($7.50)
___ LS050F84 DECSYSTEM 10/20-VAX INTEGRATION QUESTION AND ANSWER ($15.00)
___ LS051F84 VMS SYSTEM MANAGEMENT OVERVIEW FOR DECSYSTEM MANAGERS ($7.50)
___ LS053F84 DECSYSTEM 10/20-VAX INTEGRATION PANEL ($7.50)
___ LS056F84 DECSYSTEM-10 AND DECSYSTEM/20 TO VAX NETWORKING WORKSHOP ($7.50)
___ LS057F84 DIGITAL'S INTEGRATION STRATEGY FOR TOPS CUSTOMERS ($7.50)
___ LS058F84 VMS FOR DECSYSTEM-10 AND DECSYSTEM-20 USERS ($7.50)
___ LS059F84 INTEGRATION TOOLS AVAILABLE FOR TOPS CUSTOMERS ($7.50)
___ LS060F84 ALTERNATIVE MIGRATION STRATEGIES FOR LARGE SYSTEMS ($7.50)
___ LS064F84 PERSONAL COMPUTERS AND DIGITAL'S LARGE SYSTEMS ($7.50)

LANGUAGES AND TOOLS SIG

___ LT001F84 CONVERSION & COMPARISON OF STANDARD FORTRAN, DEC FORTRAN'S & VAX-11 FORTRAN ($7.50)
___ LT002F84 USING DCL AS A SOFTWARE DEVELOPMENT TOOL ($7.50)
___ LT003F84 THE USE OF DEC/CMS & DEC/MMS IN CONFIGURATION MANAGEMENT ($7.50)
___ LT006F84 MULTI-LINGUAL APPLICATIONS DEVELOPMENT ($7.50)
___ LT007F84 WRITING EFFICIENT AND EASY-TO-READ CODE IN VAX C ($7.50)
___ LT008F84 TEX: TYPESETTING FOR ALMOST EVERYBODY ($7.50)
___ LT013F84 A DOD ADA STATUS REPORT ($7.50)
___ LT014F84 PASCAL PROGRAM DEVELOPMENT REQUIRING ACCESS TO RSX-LLM EXECUTIVE DIRECTIVES ($7.50)
___ LT015F84 LANGUAGES & TOOLS QUESTION AND ANSWER ($7.50)
___ LT016F84 LANGUAGES & TOOLS PRESENTS ADA KEYNOTE ($7.50)
___ LT017F84 PROMOTING THE ACQUISITION AND USE OF SOFTWARE DEVELOPMENT TOOLS ($7.50)
___ LT018F84 REAL WORLD USE OF SOFTARE DEVELOPMENT TOOLS ($7.50)
___ LT019F84 A LANGUAGE FOR PROCESSING TEXT ON VMS ($7.50)
___ LT021F84 MAT, A STATIC ANALYZER FOR SETS OF FORTRAN MODULES ($7.50)
___ LT022F84 A STANDARD FOR SOFTWARE VERIFICATION PLANS ($7.50)
___ LT025F84 HOW TO GET THE MOST OUT OF DEC/MMS ($7.50)
___ LT026F84 WHAT'S NEW WITH DEC/CMS? CAPABILITIES FEATURES ($7.50)
___ LT027F84 DEC/CMS, DEC/MMS, DEC/TEST MANAGER-A TUTORIAL ON VMS TOOLS ($7.50)
___ LT028F84 FORTRAN IN THE VAX PROGRAMMING ENVIRONMENT ($7.50)
___ LT029F84 A NEW EMPIRICAL ANALYSIS OF FORTRAN PROGRAMS ($7.50)
___ LT030F84 VAX PL/I OVERVIEW ($7.50)
___ LT031F84 PROGRAM EDITING IN THE VMS ENVIRONMENT ($7.50)
___ LT032F84 UNIX EMULATION USING THE VAX C RUN TIME LIBRARY ($7.50)
___ LT033F84 OVERVIEW OF DIGITAL'S ADA (R) LANGUAGES IMPLEMENTATION FOR VAX/VMS ($7.50)
___ LT034F84 VAX C AND THE VMS PROGRAMMING ENVIRONMENT ($7.50)
___ LT035F84 LANGUAGE SENSITIVE EDITORS ($7.50)
___ LT036F84 VAX ADA (R) TECHNICAL SESSION ($7.50)
___ LT037F84 ANALYZING PROGRAM PERFORMANCE IN THE VMS ENVIRONMENT ($7.50)
___ LT039F84 VAX PASCAL OPTIMIZATIONS ($7.50)
___ LT040F84 VAX PASCAL FUTURES ($7.50)
___ LT041F84 CAPABILITIES OF DEC/TEST MANAGER-REGRESSION TEST TOOL FOR SOFTWARE DEVELOPMENT AND MAINTENANCE ($7.50)
___ LT042F84 DEBUGGING PDP-11 FORTRAN-77 APPLICATIONS USING FORTRAN-77 DEBUG ($7.50)
___ LT043F84 PRACTICAL BENEFITS OF A STANDARD DEVELOPMENT ENVIRONMENT ON DEC OPERATING SYSTEMS ($7.50)
___ LT044F84 CALLING PDP-11 SORT/MERGE V3.0 FROM HIGH LEVEL LANGUAGES ($7.50)
___ LT045F84 PDP-11 SORT/MERGE V3.0 ($7.50)
___ LT046F84 ADVANCED OBJECT LIBRARY CONTROL TECHNIQUES ($7.50)
___ LT047F84 INFORMAL LIBRARY CONTROL SYSTEM ($7.50)
___ LT048F84 LANGUAGES & TOOLS WRAPUP ($7.50)
___ LT049F84 LANGUAGES & TOOLS ROADMAP ($7.50)
___ LT050F84 THE USES OF TOOLS TO MIGRATE AND CONTROL CODE FROM CMS TO UNIX ($7.50)
___ LT052F84 A COMPARISON OF SEVERAL POPULAR TEXT EDITORS ($7.50)

___ LT056F84 PRODUCTIVITY ON VMS: DEVELOPMENT AT DIGITAL USING SOFTWARE TOOLS ($7.50)
___ LT057F84 TEAMS OF TOOLS — HOW TO USE DEC/CMS, DEC/MMS, DEC/TEST MANAGER ($7.50)
___ LT058F84 EMACS: AN EDITING ENVIRONMENT FOR VMS ($7.50)
___ LT059F84 FORTRAN-77 AND RT-11 OR HOW TO IMPROVE YOUR CHARACTER ($7.50)
___ LT060F84 FORTRAN-77/RT-11 RUN-TIME INTERNALS ($7.50)

MUMPS SIG

___ M002F84 DSM-11 FILE STRUCTURE ($7.50)
___ M004F84 HIGH PERFORMANCE DATABASE SYSTEM FOR VAX OR PDP-11 ($7.50)

NETWORKS SIG

___ N001F84 NETWORKS AND COMMUNICATIONS UPDATE ($7.50)
___ N002F84 DIGITAL'S DISTRIBUTED COMPUTER NETWORK ($7.50)
___ N003F84 NETWORK COMMUNICATION HARDWARE ($7.50)
___ N004F84 TRANSPORT PROTOCOL COMPARISON ($7.50)
___ N005F84 DEC, IBM, DDP...THE COMMITMENT CONTINUES ($7.50)
___ N006F84 DECNET TUTORIAL ($7.50)
___ N007F84 DIGITAL AND PACKETSWITCHED NETWORKS ($7.50)
___ N008F84 DECNET/SNA APLICATIONS DEVELOPMENT ($7.50)
___ N009F84 NETWORK PERFORMANCE ($7.50)
___ N012F84 ACCESS METHODS FOR LAN'S ($7.50)
___ N015F84 USING BROADBAND WITH DIGITAL'S NETWORK PRODUCTS ($7.50)
___ N016F84 TERMINAL SERVER FOR ETHERNET LAN'S ($7.50)
___ N017F84 ENGINEERING TRADEOFFS FOR ETHERNET CONFIGURATIONS ($7.50)
___ N018F84 TERMINAL COMMUNICATION ALTERNATIVES ($7.50)
___ N019F84 DECNET FOR THE PROFESSIONAL PERSONAL COMPUTER ($7.50)
___ N020F84 COMMUNICATION HARDWARE Q&A ($7.50)
___ N023F84 DECNET REMOTE FILE ACCESS FOR MACRO-11 PROGRAMS ($7.50)
___ N024F84 ETHERNET IN THE MANUFACTURING ENVIRONMENT, TWO ARCHITECTURES ($7.50)
___ N025F84 NETWORK MECHANICS: A NUTS AND BOLTS PRIMER ($7.50)
___ N027F84 NETWORKING STRATEGIES AT SIERRA SEMICONDUCTOR ($7.50)
___ N028F84 DIGITAL'S NETWORK MANAGEMENT SERVICES ($7.50)
___ N029F84 AN EXAMPLE OF A TOKEN PASS NETWORK WITH BUS ARCHITECTURE ($7.50)
___ N030F84 NETWORKS APPLICATION PANEL ($7.50)
___ N031F84 NETWORK EVENT LOGGING UNDER VAX/VMS ($7.50)
___ N032F84 VMS REMOTE SYSTEM AND NETWORK MANAGEMENT ($7.50)
___ N033F84 NETWORKING STANDARDS ($7.50)
___ N036F84 FACTORY AUTOMATION FOR SEMICONDUCTOR ASSEMBLY USING RDB AND ELAN ($7.50)
___ N037F84 DDCMP INTERNALS TUTORIAL ($15.00)
___ N038F84 WRITING AN APPLICATION FOR THE DECNET/SNA GATEWAY ($7.50)
___ N039F84 REAL-TIME DATA ACQUISITION AND CONTROL USING ETHERNET ($7.50)
___ N040F84 ANALYSIS OF PACKET SWITCHED COMMUNICAATION NETWORKS ($7.50)
___ N041F84 DEVELOPMENT OF A NETWORK FILE MANAGEMENT SYSTEM ($7.50)
___ N042F84 DEVELOPMENT OF A NETWORK VIRTUAL TERMINAL SERVICE ($7.50)
___ N043F84 OFFLOADING YOUR HOST USING PROGRAMMABLE FRONT END PROCESSORS ($7.50)
___ N044F84 THE BOSTON UNIVERSITY CAMPUS NETWORK: PLANS AND IMPLEMENTATION ($7.50)
___ N045F84 UNIBUS COMMUNICATION CONTROLLER ($7.50)
___ N046F84 SITE NODE IMPLEMENTATION ($7.50)
___ N047F84 VAX BASED FILE SERVERS ($7.50)
___ N050F84 PROGRAMMABLE FRONT-END PROCESSORS FOR VAX AND PDP-11 SYSTEMS ($7.50)
___ N053F84 NETWORKS Q&A ($7.50)
___ N054F84 THE NYC FIRE DEPARTMENT'S STARFIRE DISPATCHING SYSTEM ($7.50)
___ N055F84 MANUFACTURING AUTOMATION PROTOCOL ($7.50)
___ N056F84 DECLUSTERS: USING N-E-TALK TO MAKE MULTIPLE SYSTEMS APPEAR AS ONE ($7.50)

OFFICE AUTOMATION SIG

___ O001F84 OFFICE AUTOMATION SIG ROADMAP ($7.50)
___ O003F84 ALL-IN-1 QUESTION AND ANSWER PANEL ($15.00)
___ O005F84 MICROCOMPUTERS IN THE WORKPLACE ($7.50)
___ O006F84 TRAINING TECHNIQUES TO INSURE SUCCESSFUL OFFICE AUTOMATION IMPLEMENTATION ($7.50)
___ O010F84 THE DECPAGE APPROACH TO INTEGRATING AND PRINTING TEXT AND GRAPHICS IN THE OFFICE ($7.50)
___ O011F84 CUSTOMIZING PRO/OFFICE WORKSTATION-EXPERIENCES AND EXAMPLES ($7.50)
___ O012F84 PRO/OFFICE WORKSTATION INTERNALS ($7.50)
___ O013F84 VAX DECALC UPDATE AND OVERVIEW ($7.50)
___ O015F84 MANAGING THE INFORMATION SPECTRUM ($7.50)
___ O017F84 ALL-IN-1 UPDATE ($7.50)
___ O018F84 ALL-IN-1 AND 3RD PARTY APPLICATIONS — PLANNING FOR INTEGRATION ($7.50)
___ O019F84 HOW TO INTEGRATE APPLICATIONS INTO ALL-IN-1-A TECHNICAL PERSPECTIVE ($7.50)
___ O020F84 ALL-IN-1 ARCHITECTURE: A TECHNICAL OVERVIEW ($7.50)
___ O021F84 ALL-IN-1 PERFORMANCE: PLANNING AND MANAGING FOR SUCCESS ($7.50)
___ O023F84 ALL-IN-1 WORKLOAD CHARACTERIZATION OF ALL-IN-1 SYSTEMS ($7.50)
___ O024F84 WORD PROCESSING APPLICATION PERFORMANCE UNDER ALL-IN-1 ($7.50)
___ O026F84 OFFICE WORKSTATION PRODUCTS AND POSITIONING ($7.50)
___ O027F84 DOCUMENT PROCESSING PRODUCTS UPDATE ($7.50)
___ O028F84 WPS-8/DECMATE SOFTWARE UPDATE ($7.50)
___ O030F84 DEC OFFICE SYSTEM STRATEGY ($7.50)
___ O033F84 DECMATE SYSTEMS UPDATE ($7.50)
___ O034F84 WPS-PLUS TECHNICAL UPDATE ($7.50)
___ O035F84 DECDX PRODUCTS ADN DX FORMATS ($7.50)
___ O036F84 ELECTRONIC MAIL PRODUCTS UPDATE ($7.50)
___ O040F84 IBM TO DEC CONNECTIONS FOR OFFICE-GENERAL OVERVIEW OF PRODUCTS FOR INTERCONNECTION ($7.50)
___ O045F84 ALL-IN-1 IN A NETWORKED ENVIRONMENT ($7.50)
___ O046F84 RAINBOW WORKSTATION TECHICAL UPDATE ($7.50)

___ O047F84 INTERNATIONAL STANDARDS — WHAT IS DIGITAL DOING WITH THEM IN ITS OFFICE PRODUCT ($7.50)

___ O048F84 OA SIG KEYNOTE-OFFICE AUTOMATION, WHAT WE'VE LEARNED ($7.50)

___ O050F84 SUCCESSFUL OFFICE SYSTEMS IMPLEMENTATION — A PANEL ON SUPPORT REQUIREMENTS/ISSUES ($7.50)

___ O052F84 3M'S TRAC OFFICE SYSTEM — WHAT WE DID AND WHAT WE LEARNED ($7.50)

___ O053F84 DIGITAL'S COMPOUND DOCUMENT ARCHITECTURE ($7.50)

___ O055F84 ANALYZING OFFICE AUTOMATION REQUIREMENTS — A MANAGEMENT PERSPECTIVE ($7.50)

___ O061F84 THE APPLICATION OF COMPUTER-BASED TELECONFERENCING TO OFFICE AUTOMATION AND ORGANIZATIONAL CHANGE ($7.50)

___ O062F84 AN ALTERNATIVE TO THE PROJECT LIFE CYCLE (PLC) ($7.50)

___ O063F84 DIGITAL'S USE OF THE BUSINESS CONSULTING APPROACH ($7.50)

___ O064F84 WORLDWIDE ELECTRONIC MAIL ($7.50)

PC SIG

___ P003F84 PERSONAL COMPUTER PRODUCT PANEL: RAINBOW, DECMATE 11, AND PROFESSIONAL SERIES ($7.50)

___ P004F84 RAINBOW TECHNICAL Q&A ($7.50)

___ P005F84 PROFESSIONAL 300 SERIES TECHNICAL Q&A ($15.00)

___ P008F84 WILL PRO-IV REPLACE COBOL? ($7.50)

___ P009F84 INTEGRATED APPLICATIONS FOR THE PROFESSIONAL 350 ($7.50)

___ P010F84 PRO 300 NEW PRODUCTS PANEL ($7.50)

___ P012F84 PRO 300 CLUSTERS — SUPPORT FOR DISKLESS WORKSTATIONS & RESOURCE SHARING ($7.50)

___ P013F84 DOCUMENT CREATION FOR THE PRO ($7.50)

___ P014F84 PRO APPLICATION DEVELOPMENT AND TUNING ($7.50)

___ P015F84 PRO COMMUNICATIONS ($7.50)

___ P016F84 THE PROFESSIONAL SERIES GRAPHICS APLICATIONS DEVELOPMENT ($7.50)

___ P017F84 COMPUTER GRAPHICS ON THE PROFESSIONAL ($7.50)

___ P018F84 P/OS INTERNALS ($7.50)

___ P019F84 RAINBOW TO VAX SUCCESS STORIES ($7.50)

___ P021F84 RAINBOW GRAPHICS SOLUTIONS ($7.50)

___ P022F84 RAINBOW COMMUNICATIONS SOLUTIONS ($7.50)

___ P023F84 RAINBOW COMMUNICATIONS TECHNICAL SESSION ($7.50)

___ P024F84 RAINBOW GRAPHICS TECHNICAL SESSION ($7.50)

___ P025F84 INTRODUCTION TO RAINBOW CP/M AND MS-DOS ($7.50)

___ P027F84 RAINBOW MS-DOS ARCHITECTURE AND INTERNALS ($7.50)

___ P028F84 RAINBOW 100 THIRD PARTY HARDWARE/SOFTWARE ($7.50)

___ P029F84 RAINBOW USE AT SOUTHERN CONNECTICUT STATE UNIVERSITY ($7.50)

___ P030F84 OVERVIEW OF PROFESSIONAL-SERIES DOCUMENTATION ($7.50)

___ P031F84 GUIDE TO RAINBOW DOCUMENTATION ($7.50)

___ P033F84 MANAGING WITH MICROS: THE RAINBOW IN A GEOGRAPHICALLY WIDESPREAD ORGANIZATION ($7.50)

___ P035F84 LATEST DCS PRODUCTS ($7.50)

___ P036F84 DIGITAL CLASSIFIED SOFTWARE AND VERTICAL APPLICATIONS ($7.50)

___ P038F84 INTEGRATED SOFTWARE PRODUCTS FROM DCS ($7.50)

___ P041F84 RAINBOW 100-ALTERNATE OPERATING SYSTEMS/ENVIRONMENTS ($7.50)

___ P042F84 CUTTING UNIX DOWN TO SIZE ($7.50)

___ P043F84 INDUSTRY STANDARD SOFTWARE IN A MULTI-USER ENVIRONMENT ($7.50)

RSTS SIG

___ RS005F84 RSTS TECHNICAL TIPS I (SPR) ($7.50)

___ RS006F84 RSTS TECHNICAL TIPS II (BITS AND BYTES) ($7.50)

___ RS007F84 RSTS TECHNICAL TIPS III (APPLICATIONS) ($7.50)

___ RS009F84 DISASTER RECOVERY PLANNING — FROM PRACTICAL EXPERIENCE ($7.50)

___ RS010F84 DECWORD/DP PRODUCT ENAHNCEMENTS ($7.50)

___ RS011F84 CREATING A WELL STRUCTURED RSTS RDS 1 DISK ($7.50)

___ RS012F84 RSTS/E MAGIC: ADVANCED TIPS AND TECHNIQUES ($15.00)

___ RS015F84 OFFICE APPLICATIONS FOR RSTS SYSTEMS ($7.50)

___ RS016F84 RSTS/E ANNOUNCEMENTS AND FUTURES ($7.50)

___ RS017F84 FUTURE RSTS HARDWARE SUPPORT ($7.50)

___ RS018F84 ASYNCHRONOUS I/O IN RSTS SYSTEMS ($7.50)

___ RS019F84 RSTS/E LAYERED PRODUCT PANEL ($7.50)

___ RS020F84 RSTS/E: AN OVERVIEW OF V9.0 ($15.00)

___ RS021F84 ACCOUNT MANAGEMENT FOR RSTS VERSION 9.0 ($7.50)

___ RS022F84 DCL ENHANCEMENTS FOR RSTS VERSION 9.0 ($7.50)

___ RS023F84 MICRO/RSTS V1.1 AND BEYOND ($7.50)

___ RS026F84 CACHING WELL ON RSTS ($7.50)

___ RS029F84 RMS-11 TUNING ($7.50)

___ RS030F84 DECNET/E V2.1 OVERVIEW ($7.50)

___ RS031F84 NEW FEATURES AND DIRECTIVES INRSTS/E V9.0 ($7.50)

___ RS032F84 RSTS/E DIRECTORY STRUCTURE ($7.50)

___ RS033F84 RSTS/E V9.0 INSTALLATION PROCESS ($7.50)

___ RS034F84 NEW BACK UP UTILITY FOR V9.0 OF RSTS/E ($7.50)

___ RS035F84 PDP-11 HARDWARE PLANNING FOR BUSINESS USERS ($7.50)

___ RS036F84 RS232 MAGIC-MODEMS AND CABLES ($7.50)

RT-11 SIG

___ RT002F84 RT-11 USER APPLICATION WORKSHOP ($7.50)

___ RT005F84 RT-11 USER SPEAKOUT ($15.00)

___ RT006F84 ODTVO9: RT-11 SYMBOLIC DEBUGGING TOOL ($7.50)

___ RT007F84 RT-11 ZOOM DIGITAL SIGNAL PROCESSING PROGRAM FOR A PDP-11/34 ($7.50)

___ RT009F84 RT-11 USER COMMAND LINKAGE ($7.50)

___ RT010F84 THE EQ HANDLER: DATA ACQUISITION/CONTROL WITH A COPY COMMAND ($7.50)

___ RT012F84 USING THE KXT11-C I/O PROCESSOR WITH RT-11 ($7.50)

___ RT013F84 PRO-350 GRAPHICS UNDER RT-11 ($7.50)

___ RT019F84 FREE RT-11 COMMUNICATIONS PROGRAMS ($7.50)

___ RT020F84 THE RT-11/TSX OFFICE ($7.50)

___ RT021F84 RTTS/E RT-11 RUN-TIME SYSTEM ($7.50)

___ RT022F84 RT-11 USER FEEDBACK ($7.50)

RSX-11 SIG

___ RX006F84 THE LORE OF RSX ($15.00)

___ RX008F84 RSX PRODUCT PANEL ($7.50)

___ RX014F84 TASK OVERLAY STRUCTURES ($7.50)

___ RX015F84 UNUSUAL RSX DEVICE DRIVERS AND THEIR USES ($7.50)

___ RX016F84 A HAMSTER THEORY OF THE RSX-11 OPERATING SYSTEM ($7.50)

___ RX017F84 DISASTER RECOVERY: AN UNUSAL DISASTER AND A CINDERELLA RECOVERY ($7.50)

___ RX018F84 JUST A RSX MODEST PROPOSAL ($7.50)

___ RX020F84 NIFTY THINGS TO DO WITH RSX INDIRECT COMMAND FILES ($7.50)

___ RX021F84 RSX NOVICE Q/A ($7.50)

___ RX022F84 RSX ADVANCED Q/A ($7.50)

___ RX024F84 WHAT DEC ALWAYS WANTED TO KNOW ABOUT RSX BUT WAS AFRAID TO ASK ($7.50)

___ RX026F84 CONTRASTING FLAVORS OF RSX ($7.50)

___ RX027F84 RSX/16 BIT LANGUAGES PRODUCT PANEL ($7.50)

___ RX029F84 BEST OF THE SIG TAPES ($7.50)

___ RX030F84 A SIMPLE DATA ACQUISITION AND COMMUNICATION TASK USING RSX-11M ($7.50)

___ RX031F84 MIGRATION FROM RSX-11M TO 11M$ (PANEL DISCUSSION) ($7.50)

___ RX032F84 UTILITIES WALK-THROUGH FOR NOVICE RSX USERS ($7.50)

___ RX034F84 WHAT THE PDP-11 LOOKS LIKE TO RSX ($7.50)

___ RX035F84 AN INTRODUCTION TO WRITING AN RSX DEVICE DRIVER ($7.50)

___ RX036F84 RSX-11M PLUS MULTI-PROCESSING INTERNALS ($7.50)

___ RX037F84 SYSTEM MANAGER ACTIVITIES FOR 11M/M$ ($7.50)

___ RX038F84 ADVANCED FEATURES OF RSX-11M-PLUS ($7.50)

___ RX040F84 RSX SYSTEMS PROGRAMMING SHORT NOTES ($15.00)

___ RX041F84 NEW FEATURES OF RSX-11M$ V2.2 MICRO/ RSX V2.0 ($7.50)

___ RX043F84 VAX-11 RSX INTERNALS ($7.50)

___ RX044F84 USING PERFORMANCE MEASUREMENT TOOLS WITH RSX ($7.50)

___ RX045F84 COMPLEX SYSTEM DEBUGGING SINGLE TASK, MULTI-TASK, MULTI-PROCESSOR ($7.50)

___ RX047F84 THE WELL-TEMPERED CLUSTER LIBRARY-IMPROVEMENTS TO THE AUTOLOAD OVERLAY ROUTINES ($7.50)

___ RX050F84 INTRODUCTION TO RUN-OFF ($7.50)

___ RX051F84 RSX ADN MICRO RSX DOCUMENTATION OVERVIEW ($7.50)

___ RX058F84 IMPROVED RSX-11 DIRECTORY SYSTEM ($7.50)

___ RX059F84 QD: A RSX DISTRIBUTED FILE SYSTEM ($7.50)

SITE MANAGEMENT AND TRAINING SIG

___ S005F84 DON'T GET BURNED! AN OVERVIEW OF COMPUTER ROOM FIRE PROTECTION ($7.50)

___ S006F84 REAL WORLD DISK COMPARISONS ($7.50)

___ S008F84 COMPUTER SECURITY-MAKING AND BREAKING IT ($7.50)

___ S010F84 MICRO-BASED MONITOR FOR VMS SYSTEMS ($7.50)

___ S011F84 DEVELOPING A CONFIGURATION MANAGEMENT SYSTEM ($7.50)

___ S012F84 A DISK SYSTEM UPGRADE CASE STUDY ($7.50)

___ S013F84 RISK MANAGEMENT PLANNING — WHY AND HOW ($7.50)

___ S014F84 AN OVERVIEW OF COMPUTER SITE PLANNING WITH Q&A ($7.50)

___ S016F84 FIELD SERVICE ANSWERS YOUR QUESTIONS ($7.50)

___ S018F84 POWER PROBLEMS, CAUSES AND SOLUTIONS ($7.50)

___ S019F84 HUMAN FACTORS IN DESIGN ($7.50)

___ S020F84 ADD-ON HANDS-ON: UNIBUS INSTALLATION BASICS ($7.50)

___ S021F84 USER CONSULTANTS: KNOWLEDGE TRANSFER IN A MULTI-VAX COMMUNITY ($7.50)

___ S022F84 FORECASTING SYSTEM UTILIZATION — A DISCUSSION OF ONE APPROACH ($7.50)

___ S023F84 RELOCATING A COMPUTER CENTER — PLANNING AND COMMUNICATION MAKE THE DIFFERENCE ($7.50)

___ S024F84 WRITING USER FRIENDLY DOCUMENTATION ($7.50)

___ S026F84 DEVELOPING THE CAD/CAM ORGANIZAITON IN THE 80'S ($7.50)

___ S031F84 DIGITAL: YOUR COMPLETE SERVICE VENDOR ($7.50)

___ S032F84 CODING AND DESIGN TECHNIQUES FOR OVERLAYED TASKS ($7.50)

___ S036F84 REAL WORLD CONSIDERATIONS IN SETTING UP A COMPUTER FACILITY ($7.50)

___ S037F84 TECHNICAL PRESENTATION ON THE DEVELOPMENT OF DIGITAL'S SOFTWARE INFORMATION NETWORK ($7.50)

___ S038F84 IMPLEMENTING NEW TRAINING TECHNOLOGIES IN YOUR ENVIRONMENT ($7.50)

___ S040F84 OPTIMIZING LEARNING THROUGH SELF-PACED INSTRUCTION ($7.50)

___ S041F84 VAX PRODUCER, DIGITAL'S MOST ADVANCED AUTHORING SYSTEM: FORM VT 100 TO IVIS ($7.50)

___ S042F84 COURSEWARE DEVELOPMENT PROCESS ($7.50)

___ S043F84 "CLEANING UP YOUR ELECTRICAL POWER" ($7.50)

___ S044F84 THE IMPACT OF INFORMATION CENTERS ON END USER COMPUTING ($7.50)

___ S045F84 TEKTRONIX'S APPLICATIONS OF DIGITAL SELF-MAINTENANCE: A CASE STUDY ($7.50)

___ S046F84 SOFTWARE PRODUCT SERVICES PANEL ($7.50)

___ S047F84 DIGITAL SOFTWARE LICENSING OVERVIEW ($7.50)

UNISIG SIG

___ U008F84 NEW 1/2 INCH TAPE OPTIONS AND TRADE OFFS FOR UNIX ON DEC VAX PROCESSORS ($7.50)

___ U009F84 SUPPORTING BINARY UNIXES ON MANY DIVERSE H/W CONFIGURATIONS ($7.50)

___ U010F84 UNIX PROGRAMS UNDER VMS ($7.50)

___ U012F84 EXECUTING RT-11 PROGRAMS UNDER UNIX ($7.50)

___ U013F84 SYSTEM V vs. 4.2: A COMPARISON ($7.50)

___ U014F84 SOFTWARE MIGRATION — UNIX TO VMS (A CASE HISTORY) ($7.50)

___ U015F84 A COMPARISON OF THE UNIX AND VMS OPERATING SYSTEMS ($7.50)

___ U016F84 UNIX: OVERVIEW AND PHILOSOPHY ($7.50)

___ U017F84 THE RCMP ALWAYS GETS ITS MAN ($7.50)

___ U018F84 UNIX: WHERE DID IT COME FROM, WHERE IS IT NOW, HOW DID IT GET HERE, WHERE IS IT GOING? AND SHOULD IT? ($7.50)

___ U019F84 SHELL FOR UNIX USERS ($7.50)

___ U020F84 SO WHAT'S SO GREAT ABOUT UNIX EDITORS? ($7.50)

___ U021F84 WHYS AND WHEREFORES OF THE STREAM EDITOR ($7.50)

___ U022F84 UNIX PRODUCT PANEL ($7.50)

___ U023F84 INTRODUCTION TO ULTRIX COMMUNICATIONS ($7.50)

___ U024F84 ULTRIX Q AND A ($7.50)

___ U025F84 C PROGRAMMING ACROSS ARCHITECTURES ($7.50)

___ U026F84 ULTRIX-11 A TECHNICAL OVERVIEW ($7.50)

___ U027F84 C SHELL TUTORIAL ($7.50)

___ U028F84 ULTRIX FEEDBACK SESSION ($7.50)

___ U030F84 UNIX TEXT PROCESSING I-NROFF FOR THE BEGINNER ($7.50)

___ U031F84 UNIX TEXT PROCESSING II-TROFF FOR THE NROFFER ($7.50)

___ U032F84 UNIX TEXT PROCESSING III-NROFF AND TROFF MACRO WRITING ($7.50)

VAX SYSTEMS SIG

___ V004F84 VAX/VMS NOVICE RMS AND DISK TUNING ($7.50)

___ V005F84 VAX SYSTEMS SIG COMERCIAL WORKING GROUP MEETING ($7.50)

___ V007F84 VAX MAGIC, WAR STORIES, AND HORROR TALES ($15.00)

___ V009F84 VAX SYSTEMS ADVANCED QUESTION AND ANSWER ($15.00)

___ V012F84 SECURITY TIPS FOR THE NEW VMS SYSTEMS MANAGER ($7.50)

___ V013F84 VMS SYSGEN PARAMETERS ($7.50)

___ V014F84 VAX NOVICE QUESTIONS AND ANSWERS ($7.50)

___ V016F84 VAX SYSTEMS SIG SYSTEM IMPROVEMENT REQUESTS ($7.50)

___ V017F84 INTRODUCTION TO VMS COMMAND PROCEDURES ($7.50)

___ V018F84 NOVICE REAL-TIME PROCESSING USING VAXELN ($7.50)

___ V020F84 NOVICE REAL-TIME PROCESSING USING VAX/VMS ($7.50)

___ V021F84 HOW I MANAGE MY VAX SYSTEM (PANEL) ($7.50)

___ V022F84 VAX SITE PREPARATION ($7.50)

___ V031F84 VAX HARDWARE PANEL ($7.50)

___ V032F84 SIGNIFICANT VAX SYSGEN PARAMETERS DEFINITIONS AND FINE TUNING ($15.00)

___ V033F84 ANALYSIS AND FINE TUNING OF VAX RMS FILES FOR USAGE BY HIGHER LEVEL LANGUAGES ($7.50)

___ V034F84 A FAST INTER-PROCESS COMMUNICATION FACILITY FOR VMS ($7.50)

___ V037F84 NIFTY THINGS TO DO WITH VMS DCL COMMAND PROCEDURES ($7.50)

___ V039F84 TAPECOPY USER'S FORUM ($7.50)

___ V041F84 AN EXAMPLE IMPLEMENTATION OF A DECNET REMOTE TASK ($7.50)

___ V045F84 VAXCLUSTER USER INSTALLATION PANEL ($7.50)

___ V047F84 CLUSTER MANAGEMENT PANEL ($7.50)

___ V051F84 MEASURING AND IMPROVING PROGRAM QUALITY USING A SYSTEM SERVICE MONITOR IN VAX/VMS ($7.50)

___ V056F84 THE VAX GENERATION: MANAGEMENT DECISIONS ($7.50)

___ V057F84 NOVICE EXCEPTION HANDLING IN VMS ($7.50)

___ V058F84 USER FRIENDLY A PROGRAMMERS VIEW ($7.50)

___ V061F84 HSC50 OPERATIONS IN A VAXCLUSTER ($7.50)

___ V062F84 VAX MIGRATION/HOST-DEVELOPMENT WORKSHOP ($7.50)

___ V064F84 VAX TIMING CHALLENGES: UNIBUS IDIO "SYNC" RASIES ($7.50)

___ V065F84 WINNING CAPACITY PLANNING STRATEGIES IN A MULTIVAX FACILITY ($7.50)

___ V068F84 BETTER DCL PROGRAMMING UNDER VMS ($7.50)

___ V070F84 FOREIGN TERMINAL SUPPORT IN THE V4.0 RUN TIME LIBRARY ($7.50)

___ V071F84 SCREEN MANAGEMENT IN THE VMS RUN TIME LIBRARY ($7.50)

___ V072F84 OVERVIEW OF THE VMS RUN TIME LIBRARY FOR V4.0 ($7.50)

___ V073F84 A VMS DEVICE DRIVER IMPLEMENTING A NETWORK PROTOCOL FOR THE DR-11-W ($7.50)

___ V074F84 VAX PROFESSIONAL WORKSTATION GOES GRAPHIC ($7.50)

___ V075F84 MORE FROM YOUR MONITOR UTILITY ($7.50)

___ V076F84 VMS SYSTEM PERFORMANCE MANAGEMENT FOR NOVICES ($7.50)

___ V077F84 VAX REMOTE PRINT, PLOT, AND DOWNLOAD SOFTWARE ($7.50)

___ V078F84 GETTING ACQUAINTED WITH VAX/VMS AND MICRO VMS DOCUMENTATION ($7.50)

___ V079F84 VAX ASSEMBLY CODE OPTIMIZATIONS AND INSTRUCTION TIMING COMPARISONS ($7.50)

___ V081F84 VAX DEBUG TUTORIAL: HOW DO YOU FIND THAT BUG? ($7.50)

___ V083F84 CREATING A WELL STRUCTURED VMS ODS2 DISK ($7.50)

___ V084F84 DEC/SHELL FOR VMS USERS ($7.50)

___ V085F84 A HIGH PERFORMANCE MULTIPROCESSOR DATABASE IMPLEMENTATION FOR VAX/VMS ($7.50)

___ V086F84 WRITING VAX/VMS DEVICE DRIVERS, ACPS, AND OTHER EXECUTIVE SOFTWARE IN BLISS ($7.50)

___ V08784 MICROVMS-VMS SUPPORT FOR THE MICROVMS PRODUCT SET ($7.50)

___ V088F84 VAXCLUSTER TECHNICAL CONCEPTS ($7.50)

___ V089F84 VAXCLUSTERS — FROM THE CLUSTER MANAGEMENT VIEWPOINT ($7.50)

___ V090F84 VAX/VMS VOLUME SHADOWING ($7.50)

___ V091F84 VMS PERFORAMNCE IN A VAXCLUSTER ($7.50)

___ V092F84 VMS V4 PERFORMANCE ($7.50)

___ V093F84 DECNET-VAX PERFORMANCE ($7.50)

___ V094F84 REGRESSION TESTING OF VMS ($7.50)

___ V095F84 ENHANCEMENTS TO VAX-11 RMS FOR VMS V4 ($7.50)

___ V096F84 VAX/VMS PACKAGING AND DISTRIBUTION ($7.50)

___ V097F84 VAX/VMS UTILITIES ENHANCEMENTS IN VERSION 4.0 ($7.50)

___ V098F84 FILE NAMES, LOGICAL NAMES AND SEARCH LISTS IN VMS V4 ($7.50)

___ V099F84 DCL AND THE COMMAND DEFINITION UTILITY ($7.50)

___ V100F84 SPSS GRAPHICS: AN OVERVIEW OF DATA REPRESENTATION GRAPHICS ($7.50)

___ V101F84 VMS SECURITY FEATURES ($7.50)

___ V102F84 FUTURE DIRECTIONS IN VMS SECURITY ($7.50)

___ V103F84 DECNET-VAX TUTORIAL ($7.50)

___ V105F84 VMS PROCESSOR SUPPORT ($7.50)

___ V106F84 VMS TERMINAL DRIVER ($7.50)

___ V107F84 TRAFFIC FLOW MANAGEMENT FOR AIRPORTS ($7.50)

___ V108F84 VMS V4 BATCH AND PRINT ($7.50)

___ V109F84 VAX WORKSTATIONS — GKS INTERFACE ($7.50)

___ V110F84 VMS SOFTWARE IN A WORKSTATION ENVIRONMENT ($7.50)

___ V112F84 VAX WORKSTATIONS-HUMAN INTERFACE ($7.50)

___ V113F84 VAX WORKSTATIONS-PHILOSOPHY AND CONCEPTS ($7.50)

___ V115F84 VMS SYSTEM UPDATE ($7.50)

___ V117F84 NEW VMS V4.0 HINTS AND KINKS ($7.50)

___ V118F84 VAX COMPUTER FACILITY MANAGEMENT ($7.50)

___ V119F84 VAX RESPONSE TIME MEASUREMENT ($7.50)

___ V120F84 VAX MEMORY PERFORMANCE ISSUES ($7.50)

___ V121F84 AT LAST: A HIGH PERFORMANCE TCP/IP IMPLEMENTATION FOR VAX/VMS ($7.50)

___ V122F84 CONTROLLER BASED SHADOWING ($7.50)
___ V123F84 DISK PERFORMANCE TESTING ON VAX AND PDP-11/70 ($7.50)
___ V124F84 TAPE SERVICES ON AN HSC50 ($7.50)
___ V126F84 VAXELN PERFORMANCE ($7.50)
___ V127F84 VAXELN UPDATE AND PLANS ($7.50)
___ V128F84 VAXELN INTERNALS ($15.00)
___ V129F84 VAXELN Q&A ($7.50)
___ V130F84 THE IMPLEMENTATION OF A DEC BASED HI TECHNOLOGY CAM SYSTEM IN THE SEMICONDUCTOR ENVIRONMENT ($7.50)
___ V131F84 MICROVAX HARDWARE UPDATE ($7.50)
___ V132F84 A GENERIC OPERATING SYSTEM INTERFACE FOR VMS ($7.50)
___ V133F84 "VAX SPM-A SOFTWARE PERFORMANCE MONITOR FOR VMS ($7.50)
___ V134F84 "VAX SPM TUTORIAL-USING SPM FOR SYSTEM TUNING" ($7.50)
___ V135F84 "VAX SPM TUTORIAL-USING SPM FOR CAPACITY PLANNING" ($7.50)
___ V136F84 MIGRATION TO VAX CLUSTERS ($7.50)
___ V137F84 FRONT ENDING A VAX WITH LSI-11/23'S ($7.50)
___ V138F84 DIGITAL'S ARTIFICIAL INTELLIGENCE PRODUCTS ($7.50)
___ V141F84 OPTIMIZING LARGE VMS SYSTEMS ($7.50)
___ V142F84 INTERFACING FOREIGN ADAPTERS IN VMS ($7.50)
___ V145F84 A Q-BUS REVIEW FOR THE VAX USER ($7.50)
___ V146F84 LARGE VAX PERFORMANCE ($7.50)
___ V147F84 LARGE VAX TECHNOLOGIES ($7.50)
___ V148F84 LOCAL AREA SYSTEM CONCEPTS ($7.50)
___ V149F84 SECURE OPERATING SYSTEMS FOR VAX ($7.50)
___ V150F84 VAX UPDATE ($7.50)
___ V152F84 NON-DISCRETIONARY SECURITY FEATURES IN VMS ($7.50)
___ V153F84 VAX CLUSTER TROUBLESHOOTING ($7.50)
___ V154F84 VAX LANGUAGES AND PROGRAM SUPPORT ENVIRONMENT OVERVIEW ($7.50)
___ V156F84 VAX DEBUGGER V4.0 ($7.50)

___ 9999 **Complete set 1984 DECUS Fall Cassettes** — ask for information on the special discounted price available!

ACCESSORIES

___ CUSTOM 12-CASSETTE STORAGE ALBUM ($6.00)
___ ALLSOP CASSETTE DECK CLEANER ($7.50)
___ RQ-355 PANASONIC CASSETTE RECORDER/PLAYER ($89.95)

___ Total Individual Cassettes@$7.50 each = $ ___

___ Total 12-Cassette Specials@$82.50 each = $ ___
(includes 1 FREE cassette & 1 FREE storage album!)

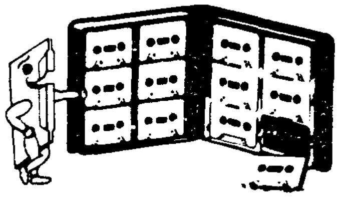
___ Other: ___________________ = $ ___
(qty.) (please specify item & price)

Maryland residents add 5% state sales tax = $ ___

Postage & Handling = $ 2.50

GRAND TOTAL = $ ___

SPECIAL VALUE



Your choice of any 12 DECUS cassettes, handsomely packaged in a custom shortage album Only **$82.50!**

(offer includes 1 **FREE** cassette & 1 **FREE** storage album!)

TO ORDER CASSETTES

Simply check the sessions you wish to purchase on this form. Cassettes are priced at $7.50 each — some sessions are presented on more than one cassette. Be sure to take advantage of the Cassette Special! Please include $2.50 to cover Postage & Handling. Fill in the GRAND TOTAL space and the information below. Send completed form, with payment, to: **Eastern Audio Associates, Inc.; 6330 Howard Lane; Elkridge, Maryland 21227; or, for credit card orders, phone (301) 796-0040.**

□ **PAID** - □ MONEY ORDER □ CHECK □ CREDIT CARD

□ AMEX □ VISA □ MASTERCARD

CREDIT CARD NUMBER

CARDHOLDER'S NAME EXPIRATION DATE

X

SIGNATURE (required on all charges) M.C. - 4 digit bank number

All purchases under $10.00 must be paid cash or check.

Make checks payable to **Eastern Audio Associates, Inc.**

□ PAYMENT $___________ ENCLOSED

Please fill out the information below before placing your order. Any order not picked up at the convention will be automatically sent to the address below.

PLEASE PRINT CLEARLY

Name	
C/O	
Street	
City	State
Zip	Phone

digital RT-11 -- CP/M -- MS-DOS Comparison

Comparing

RT-11

CP/M

MS-DOS

By Jim Williams, RT-11 Architect

RT-11 is a trademark of Digital Equipment Corporation
CP/M is a trademark of Digital Research, Inc.
MS-DOS is a trademark of Microsoft

digital RT-11 -- CP/M -- MS-DOS Comparison

Comparing 3 single-user operating systems,

using RT-11 as the "standard"

This will be a discussion of technical characteristics,

not of issues like pricing or market share.

Terminal I/O

.GTLIN	21/0A Get Input String
.MTATCH	
.MTDTCH	
.MTIN	21/03 Wait for Char from Aux
.MTOUT	21/04 Write Char to Aux
.MTPRNT	
.MTRCTO	
.MTSET	
.MTSTAT	

.PRINT	21/09 Print a String ($!)
.RCTRLO	
.TTYIN	21/01 Input w/^Break
.TTYIN/SPMOD$	21/07 Input w/o ^Break, No Echo
.TTYIN/SPMOD$/.SCCA	21/08 Input w/^Break, No Echo
.TTINR	21/06 Attempt Input
.TTINR	21/0B Check Input Status
.TTYOUT	21/02 Output w/^Break
.TTYOUT/.SCCA	21/06 Output w/o ^Break
.TTOUTR	
.MTATCH/Async Word	21/0B Check Input Status
.TTINR loop	21/0C Purge Input Buffer

.GTLIN	10 Get Input String
.MTATCH	
.MTDTCH	
.MTIN	
.MTOUT	
.MTPRNT	
.MTRCTO	
.MTSET	
.MTSTAT	

Terminal I/O (cont.)

.PRINT	09 Print a String ($!)
.RCTRLO	
.TTYIN	01 Input
.TTINR	06 Attempt Input
.TTINR	11 Check for Input
.TTYOUT	02 Output
.TTYOUT	06 Output
.TTOUTR	

File Open/Close Operations

.ABTIO	
.CHCOPY	
.CLOSE	21/10 Close a File
.CLOSE	21/3E Close a File
.CSIGEN	21/29 Parse a File Name
.ENTER	21/16 Create a File
.ENTER	21/3C Create a File
.LOOKUP	21/0F Open a File
.LOOKUP	21/3D Open a File
.PURGE	
.REOPEN	21/46 Point to another file
.SAVESTATUS	21/45 Another chan for file

.ABTIO	
.CHCOPY	
.CLOSE	16 Close a File
.CSIGEN	
.ENTER	22 Create a File
.ENTER	27 Get Info About Space
.LOOKUP	15 Open a File
.LOOKUP	35 Addr of Last Record
.PURGE	
.REOPEN	
.SAVESTATUS	
	17 Search for First File
	18 Search for Next File

File/Device Input Operations

.READ		
.READC		
.READW	21/14	Read a Seq. Record
.READW BUF=	21/1A	Set Disk Transfer Addr
.READW	21/21	Read a Random Record
.READW BLK=	21/24	Set Random Record Field
.READW	21/27	Read Multiple Random Records
.READW	21/3F	Read
.READW	21/44/2	Read
.READW	21/44/4	Read
.SPFUN	25	Read Sectors
.WAIT		
	21/44/6	Get Input Status

File/Device Input Operations

.READ		
.READC		
.READW	20	Read a Seq. Record
.READW BUF=	26	Set Disk Transfer Addr
.READW	33	Read a Random Record
.READW WCNT=	44	Set Record Count
.WAIT		

File/Device Output Operations

.SPFUN	26	Write Sectors
.WAIT		
.WRITC		
.WRITE		
.WRITW	21/05	Write to Printer
.WRITW BUF=	21/1A	Set Disk Transfer Addr
.WRITW	21/15	Write a Seq. Record
.WRITW	21/22	Write a Random Record
.WRITW BLK=	21/24	Set Random Record Field
.WRITW	21/28	Write Multiple Random Records

.WRITW	21/40 Write
.WRITW	21/44/3 Write
.WRITW	21/44/5 Write
	21/44/7 Get Output Status
	21/2E Read / Write Verify Status

File/Device Output Operations

.WAIT	
.WRITC	
.WRITE	
.WRITW	04 Write to Punch
.WRITW	05 Write to Printer
.WRITW BUF=	26 Set Disk Transfer Addr
.WRITW	21 Write a Seq. Record
.WRITW	34 Write a Random Record
.WRITW	40 Write Random Record W/O Fill
.WRITW WCNT=	44 Set Record Count

File Maintenance Operations

.CDFN	
.CSIGEN	21/29 Parse a File Name
.CSISPC	21/29 Parse a File Name
.CSTAT	21/23 Get File Size
	21/28/0 Set File Size
.DELETE	21/13 Delete a File
.DELETE	21/41 Delete a File
.FETCH	
	21/43 Get File Attributes
.FPROT	21/43 Set File Attributes
.QSET	
.RELEASE	
	21/11 Search for First File
	21/12 Search for Next File

File Maintenance Operations

.CDFN	Allocate Space for FDBs
.CSIGEN	
.CSISPC	
.CSTAT	7 Get Logical Assign Info
.CSTAT	25 Get Default Drive Assign
.DELETE	19 Delete a File
.FETCH	
.FPROT	30 Set File Attributes
.QSET	
.RENAME	23 Rename File
.RELEASE	
.SFDAT	
	41 Test and Write Record
	42 Lock Record
	43 Unlock Record
	46 Get Free Space
	48 Flush Buffers
	51 Set File Buf Seg Base
	52 Get File Buffer Addr
	101 Get Directory Data
	102 Read XFCB
	103 Write XFCB
	106 Set Password

Device Open/Close Operations

.CLOSE	
.DEVICE	
.DSTATUS	21/44/0 Get Device Information
$SET xx: ___	21/44/1 Set Device Information
.ENTER	
.FETCH	
.LOOKUP	
.PEEK	21/25 Get Interrupt Vector
.PROTECT	21/25 Set Interrupt Vector
.QSET	

.RELEASE	
.UNPROTECT	
$ASSIGN __: DK:	21/0E Select Default Disk
	21/0D Disk Reset
	21/54 Get Read/Write Verify Status

Device Open/Close Operations

.CLOSE	
.DEVICE	
.DSTATUS	
.ENTER	
.FETCH	
.LOOKUP	
.PROTECT	
.QSET	
.RELEASE	
.UNPROTECT	
$ASSIGN __: DK:	14 Select Default Disk
	13 Disk Reset
	24 Get Active Drive Map
	29 Get Read only Map
	31 Get Disk Parameters
	37 Reset Drive
	38 Access Drive
	39 Free Drive

Extended Memory Support

.CRAW	
.CRRG	21/48 Allocate Memory
.ELAW	
.ELRG	21/49 Free Memory
.ELRG/.CRRG	21/4A Change Allocation
.GMCX	
.MAP	
.RDBBK	
.RDBDF	
.UNMAP	
.WDBBK	
.WDBDF	

Extended Memory Support

.CRAW	
.CRRG	53 Allocate Memory
.CRRG	55 Allocate Relocatable Memory
.ELAW	
.ELRG	57 Free Memory
.ELRG	58 Free All Memory
.GMCX	
.MAP	
.RDBBK	
.RDBDF	
.UNMAP	
.WDBBK	
.WDBDF	

Date and Time

.CMKT	
.DATE	21/2A Get Date
.GTIM	21/2C Get Time
.MRKT	
.SDTTM	21/2B Set Date
.SDTTM	21/2D Set Time
.TWAIT	

Date and Time

.CMKT	
.DATE	105 Get Date
.GTIM	105 Get Time
.MRKT	
.SDTTM	104 Set Date & Time
.TWAIT	

Multi-Job Services

.CHCOPY
.CNTXSW
.GTJB
.LOCK
.MWAIT
.RCVD
.RCVDC
.RCVDW
.RSUM
.SDAT
.SDATC
.SDATW
.SPND
.TLOCK
.UNLOCK

Exit Services

.CHAIN	21/4B	Execute/Load Program
.EXIT R0≠0		
.EXIT R0=0	20	Terminate program
.EXIT R0=0	21/00	Terminate program
.EXIT USERRB	21/4C	Terminate w/Completion Code
.EXIT SPXIT$		
.EXIT CHNIF$		
.EXIT RSTRT$		

Exit Services

.CHAIN	59	Load Program
.EXIT R0≠0		
.EXIT R0=0	00	Terminate program
.EXIT USERRB		
.EXIT SPXIT$		
.EXIT CHNIF$	47	Chain to Command
.EXIT RSTRT$		

Monitor Misc.

.GVAL SYSVER	21/30 Get System Version Number
.HERR	
.HRESET	
.MFPS	
.MTPS	
.PEEK	
.POKE	
.PVAL	
.SCCA	21/33 ^Break Detection
.SERR	
.SETTOP	
.SFPA	
.SPCPS	
.SRESET	
.TRPSET	
	21/38 Get Country Dependent Data

Monitor Misc.

.HRESET	
.GVAL SYSVER	12 Get System Version Number
.HERR	45 Set Error Mode
.MFPS	
.MTPS	
.PEEK	
.POKE	
.PVAL	
.SCCA	
.SERR	45 Set Error Mode
.SETTOP	
.SETTOP #0	58 Release all Storage
.SFPA	
.SPCPS	
.SRESET	
.TRPSET	
	32 Get / Set User Code

Handler Writing Support

.ADDR
.CTIMIO
.DRAST
.DRBEG
.DRBOT
.DRDEF
.DREND
.DRFIN
.DRINS
.DRSET
.DRVTB
.FORK
.INTEN
.QELDF
.SYNCH
.TIMIO

Assembler Coding Support

.ADDR
.ASSUME
.BR
.CKXX

SYMPOSIUM TAPE INFORMATION

139 G Street, Suite 161
Davis, CA 95616
(916) 756-3291

At the Fall 84 DECUS symposium I volunteered to be the RUNOFF contact on the RT-11 Steering committee. In that capacity I intend to test each new version of RUNOFF that becomes available on the RT-11 symposium tapes, and to coordinate efforts to document and fix any bugs that are found. Watch the RT-11 Minitasker for future RUNOFF information.

The Spring 84 RT-11 symposium tape contains a significantly enhanced version of RUNOFF from Bonner Labs that is almost a complete emulation of DSR (Digital Standard Runoff). In addition to providing some very nice features that were not available in the previous version of RUNOFF for RT-11, this new version of RUNOFF offers high compatability between RUNOFF files created on a VAX with DSR and RUNOFF files created on RT-11. Unfortunately the Spring 84 Bonner Labs RUNOFF contained two bugs in the RT-11 implementation: 1) on 11/73s it would sometimes crash the system, and 2) on some systems it would not run after a certain date.

The Fall 84 RT-11 symposium tape contains a revised and improved version of Bonner Labs RUNOFF that not only has the previous bugs fixed but has a substantial number of enhancements. If you are a RUNOFF user, I strongly recommend that you start using this version of Bonner Lab RUNOFF.

I have tried the FALL 84 Bonner Lab RUNOFF on several 11/23 configurations, a micro-11 with a /73, a PRO-350 running TSX, and a PRO-380 running RT-V5.1. It performed well on all of these systems except the PRO-380. I think the problem on the 380 is tied up with the PRO/RT problem of running large jobs and is not a problem with Bonner Lab RUNOFF itself.

I have found one minor bug that has an easy workaround: if the last line of a RUNOFF file is to be printed in 'fill' mode, it does not get written out. The workaround is: Create a one line file with '.NJ', and specify this file at the end of the command string. For example, if the file FIX.RNO contains the single line '.NJ', then the runoff file 'MYFILE.RNO' would be typed out correctly by giving RUNOFF the CSI string 'TT:=MYFILE,FIX'.

If you encounter any problems with Bonner Lab RUNOFF, please send me a complete description of the problem preferably with the offending RUNOFF file. If I can verify the problem, I will work with the authors of Bonner Lab RUNOFF to see if it can be fixed by the next release.

Subject: RT-11 RUNOFF

DATE: 30-Jan-84

FROM: Robert Walraven
Multiware, Inc.

****

DECUS Symposium RT-11 SIG Tape

Fall, 1984
Anaheim, CA

Annotated Directory

****

IMPORTANT

Read the file, README.1ST, first.

```
README.1ST     14  02-Jan-85  SIG tape copy instructions
                              and new information for everyone.
```

NOTE! We are interested in maintaining the quality of the submissions to the RT SIG tape. Therefore, we welcome feedback regarding your use of these files, any bugs you find, and any bug fixes or improvements you devise. Please send any correspondence regarding the tape to:

John Crowell
Crow4ell Ltd.*   * (But not very)
145 Andanada
Los Alamos, NM 87544
(505) 662-3893

DCS - CROWELL

****

David Stagg
Dept of Pharmacology
Yale University Medical School
333 Cedar Street
New Haven, CT 06510
(203) 436-2151

This is a subdevice handler (necessary for RT-11 Version 4), used to read the .DSK files on this tape. See README.1ST for an explanation and instructions.

```
XD     .SYS     2  05-May-81  SJ/FB driver for V4
XDATCH.SAV     21  17-Apr-81  XDATCH utility
DUP    .SXD    41  21-Sep-81  DUP V4.0K patched for XD
XD     .DOC    10  16-Nov-81  Description document
XDX    .SYS     2  05-Apr-81  XM driver for V4
DUP    .COM     1  16-Mar-82  Patch file for V4 DUP.SAV

XD     .DEV    55  06-Apr-83  VIRTUAL DEVICE for rebuilding XD,
                                if necessary.
```

****

VIRTUL - Subdevice retriever for RSTS.

E.F.Beadel, Jr., Manager
CAUSE Instructional Computer Center
SUNY at Oswego
Oswego, NY 13126
(315) 341-3055

This program allows RSTS/E users to break down the subdevice files from this tape after they have been copied to disk. It has been modified by David Smith, Galileo Computer Center, to remove a few bugs and to be able to read multi-segment directories. See README.1ST for details.

VIRTUL.BAS   1 File,   43 Blocks
****

DIR2 - Annotated tape directories, part 2.

N. A. Bourgeois, Jr.
Sandia National Laboratories
Division 5238
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-8088

R. W. Barnard
Sandia National Laboratories
Division 7523
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-5115

Annotated directories of the DECUS Symposia RT-11 tapes from the Fall of 1981 through the Fall of 1984 (this symposium). Previous tapes have DIR1.DSK, which contains annotated directories from Spring 1978 through Spring 1981.

DIR2.DSK   7 Files, 285 Blocks
****

KER* - KERMIT File transfer protocol.
K11*.HEX

Brian Nelson
Computer Services, University of Toledo
2801 West Bancroft
Toledo, OH 43606
(419) 537-2841

This is release 2.23 of Kermit-11. It requires RSTS version 7.2 or later, RSX11M v4.0 or later, or RSX11M Plus version 2.0 or later, or RT11 version 4.0 or later, or P/OS version 2.0 or PRO/RT11 version 5.1

Minimum system requirements to run Kermit:

RT11 (excluding the PRO/350).
This version of Kermit will run on RT11 version 4.0 or later as long as the monitor has multiple terminal support generated. The use of this feature enables the user to access DZ11's as well as DL11's. If you are using XM then you should use K11XM, otherwise use K11RT4.

PRO/RT
For PRO/RT11 version 5.1, please note that you MUST make two modifications to the XC handler. See K11PRT.MAC for more information.

The distribution has been subdivided roughly by operating system. The subdevice files KERCM*.DSK contain documentation and files common to all operating systems. The other subdevices are operating-system specific. The distribution also contains both save (binary, executable) images and .HEX (ASCII) versions of the save images. See the installation document for information on how to create a binary from the hex file. For RSX and RSTS, the HEX files are not contained in subdevices. Please note that the allocation of specific files to the operating system-specific subdevices was done without a great deal of research - If you can't find a file, try another subdevice!

KERCM1.DSK	10 Files,	361 Blocks	(Common Files)
KERCM2.DSK	13 Files,	485 Blocks	(Common Files)
KERCM3.DSK	13 Files,	156 Blocks	(Common Files)
KERT1.DSK	12 Files,	263 Blocks	(RT Files)
KERT2.DSK	3 Files,	472 Blocks	(RT Files)
KERT3.DSK	3 Files,	476 Blocks	(RT Files)
KERST1.DSK	17 Files,	425 Blocks	(RSTS Files)
KERST2.DSK	3 Files,	389 Blocks	(RSTS Files)
K11NRS.HEX	1 File,	726 Blocks	(RSTS File)
KERSX1.DSK	14 Files,	284 Blocks	(RSX Files)
KERSX2.DSK	2 Files,	395 Blocks	(RSX Files)
K11RSX.HEX	1 File,	539 Blocks	(RSX File)
K11POS.HEX	1 File,	363 Blocks	(POS File)

****

MODEM7 - MODEM7/XMODEM Protocol for RT-11.

Dale J. Travis
14603 Smith Road
Lockport, IL 60441
(312) 972-6964

This program provides a protocol compatible with the CP/M world. It was originally submitted on the Spring, 1983, tape, but is repeated herein response to the talks at DECUS on communications software.

MODEM7.DSK 10 Files, 91 Blocks

****

HGRAF* - Plotting Package for RT-11.

Dennis V. Jensen
Ames Laboratory ISU/USDOE
310 Metallurgy
Ames, IA 50011
(515) 294-4823

This is the third DECUS release of HGraph. New features include support for REGIS-compatible terminals, such as the VT-125. An RT-11 FLECS translator is also included for reference. PLOT3D.FOR is included as a separate file.

HGRAF1.DSK 16 Files, 349 Blocks
HGRAF2.DSK   2 Files, 454 Blocks
HGRAF3.DSK   3 Files, 356 Blocks
****

DRDRV* - DR-11W Device Handler for high-speed interprocessor
CDPAK* - communications; Library for DR-11W routines.

Peter Heinicke
FermiLab
P. O. Box 500
Batavia, IL 60510
(312) 840-4670

This package allows RT-11 programs to communicate between processors using the DR-11W parallel interface. The system can also be used for communicating with RSX-11M systems. The CDPACK routines provice a library utilities for use with the DR driver.

DRDRV1.DSK 23 Files, 484 Blocks
DRDRV2.DSK 10 Files, 484 Blocks
DRDRV3.DSK   6 Files, 484 Blocks
CDPAK1.DSK 18 Files, 307 Blocks
CDPAK2.DSK 21 Files, 473 Blocks
****

RUNOF* - Bonner Lab RUNOFF.

John M. Clement
Bonner Lab, Rice University
P. O. Box 1892
Houston, TX 77401
(713) 527-4018

This is a new submission of the Bonner Lab Runoff text formatter, which when used with your favorite editor makes a complete word processor. Its syntax is almost a complete emulation of DSR (Digital Standard Runoff) and it is compatible with previous versions of Runoff.

This version allows complete control of any particular printer by means of user-definable escape sequences; variable spacing, subscripting, superscripting, and equation formatting are possible. In addition a macro facility allows text or sequences of commands to be abbreviated to a single label.

Table of contents, indexing and sub-indexing are all supported. Multiple table of contents or multiple indexing may be constructed. A variety of LAYOUT and STYLE commands are available to control the look of the pages. Table layout is simplified by right justified and 'decimal' justified tabs. Permanent margins which apply to the page headers are available.

Bugs that have been fixed since the Spring, 1984, distribution are in RNOBUG.DOC. The latest enhancements are in CHANGE.DOC. BRIEF.DOC is a brief version of the full documentation. An additional recent bug fix is given in README.2ND.

RUNOF1.DSK	42 Files,	472 Blocks
RUNOF2.DSK	10 Files,	486 Blocks
RUNOF3.DSK	40 Files,	486 Blocks
RUNOF4.DSK	31 Files,	408 Blocks
RUNOF5.DSK	104 Files,	257 Blocks

****

TSXLIB - Programmed Requests for TSX+.

N. A. Bourgeois, Jr.
NAB Software Services, Inc.
P. O. Box 20009
Albuquerque, NM 87154
(505) 298-2346

This is an update of the TSX+ system services available via programmed requests. They are in the form of a library, available to the FORTRAN programmer. The library has been updated to include all EMT's through TSX+ Version 5.1.

TSXLB1.DSK	22 Files,	224 Blocks
TSXLB2.DSK	2 Files,	348 Blocks

****

UCLPLS - User Command Language (UCL) Program.

William K. Walker
Monsanto Research Corp.
P. O. Box 32   OS-123
Miamisburg, OH 45342
(513) 865-3557

UCL+ is upward-compatible with the UCL distributed with RT-11, Version 5.1B and later. The version submitted to this tape is V07.27, an update from all previous versions. UCL+ contains a number of extensions, including chaining to additional UCL's, "run-by-name", path definition, display of command expansions, etc. Symbols are defined by entering a "symbol definition string" in the format: symbol==definition. The DISPLAY command can be used to output ASCII strings to the console or printer (handy for sneaky escape sequences). A UC "pseudo-device" handler is provided as an option which allows UCL+ to "remember" the "input-spec" part of the last UCL+ command. This text can be retrieved, at the command level, by using the "^" character in place of the argument in a subsequent command.

UCLPLS.DSK	19 Files,	408 Blocks

****

ODTV9* - Superset of ODT debugger for video terminals.

D. R. Hanson
Lockheed Aircraft Service Co.
Dept. 1-321
P. O. Box 33
Ontario, CA 91761
(714) 988-2427

ODTV09 contains the basic "DEC ODT", modified to gain entry points to subsequent routines. Almost all features of the original version are maintained. ODTV09 was written for use as an enhanced debugging tool for an RT-11 single user environment. Its basic mode allows the user to display application program symbols and their values on the system terminal. At normal "ODT" breakpoints, the displayed values are updated and the contents of the general purpose registers R0 thru R5 are displayed. The user has the option of encoding displayed values in hex, decimal or octal. ODTV09 provides a display capability of 35 user selectable symbols. VT52 cursor addressing sequences are used in the program, so the user must switch a VT100 to VT52 mode.

ODTV9A.DSK 17 Files, 345 Blocks
ODTV9B.DSK 1 Files, 342 Blocks
****

MISC - Miscellaneous goodies.

Robert Walraven
University of California
Applied Science
Davis, CA 95616
(916) 752-3300

This submission contains:
A library of digital filtering routines.

Routines for 2-to-3 conversion of binary files to/from ASCII printable files. These routines are handy if you must transmit a binary file to another system but don't have a communications package such as TRANSF, KERMIT, or MODEM7 at both ends.

Some Bessel function calculating subroutines.

A different type of random number generator.

A program for calculating the position of the sun throughout the day for FRESNO, CA. If you are not interested in FRESNO, substitute the longitude and latitude of your favorite city.

MISC.DSK 11 Files, 89 Blocks
****

BLKDIR - Bulk Directory system.

Dr. M. van Swaay
Department of Computer Science
Kansas State University
Manhattan, KS 66506
(913) 532-6350

Users of small RT-11 systems with floppy disk storage may easily accumulate 100 or more volumes of data, programs, text, etc. With such a large collection, locating individual files can become a major chore, requiring lengthy scanning of displayed or printed directories. A master directory extending over many disk volumes will allow more efficient searching. A set of TECO macros has been designed to create, maintain, and search master directory files extending over 50 or more floppy volumes. The search routine supports wildcard search patterns similar to those available in DIR. Comments may be added to each entry in the directory file.

BLKDIR.DSK 30 Files, 209 Blocks
****

DIRUTL - Utilities for manipulating directories, and other things.

Glenn A. Bever
NASA Ames/Dryden Flight Research Facility
Code OFEM
P.O. Box 273
Edwards, CA 93523
(805) 258-3311

These are a group of programs, subroutines and control files that manipulate directories. SDIR searches an RT-11 volume and its subdevices for specified filenames. Control files are included that compare directories, print directories, backup and restore directories in a format compatible with SDIR. PRH is a print utility to produce date/time stamped headers. Some programs are for use with TEKTRONIX development systems and PROLOG prom programmers.

DIRUTL.DSK 50 Files, 442 Blocks
****

TSXUTL - Utilities for TSX and RT.

Mike Marak, David Gaudine, Dr. S.J. Kubina
EMC LAB, Room CC-113
Loyola Campus, Concordia University
7141 Sherbrooke St. W.
Montreal, Canada   H4B 1R6
(514) 482-0320 ext 281

This is a collection of utility programs for controlling the TSX environment, including one to open a TSX system after it has been $SHUT, to send a CTRL-S to another terminal, to create and run a DETACHED job and to send a CTRL-C to another terminal. For RT and TSX, there is a cross-reference and overlay generator for

Fortran programs, and a program which replaces all tabs in a program by spaces.

TSXUTL.DSK 16 Files, 306 Blocks
****

F77UTL - FORTRAN-77 Utilities.

R. W. Barnard
Sandia National Laboratories
Division 7523
Albuquerque, NM 87185
(505) 844-5115

A collection of utilities written in FORTRAN-77. These include routines for converting ASCII (or character) strings to and from single- and double-precision integers, a single-character input routine (you can get a character without needing a carriage return to terminate the input), and several VT100 screen-control and formatting routines. The latter include cursor position report, two-column output, one line output, and centered output. There is also an IND file which allows incremental backups to be done almost like a VAX.

F77UTL.DSK 10 Files, 120 Blocks
****

FILUTL - Utilities for source file maintenance.

Walter L. Battaglia
YOLO EXPERT SOFTWARE
1111 Kennedy Place, Suite 4
Davis, CA 95616
(916) 758-8940

STRIP splits source files into comment and source code files. Useful for stripping comments or for starting a program documentation file. DCOLAT splits a file into multiple sub-files. COLATE is the reverse of DCOLAT: re-combines the files specified into 1 large file. ASIS (assign ISAM) reads COBOL-PLUS PRIMARY INDEX file in an ISAM file system. TTSUBS are miscellaneous terminal error messages subroutines.

FILUTL.DSK 11 Files, 168 Blocks
****

ALGOL* - ALGOL for RT-11.
APL* - APL for RT-11.
FORTH - FORTH for RT-11.
VLISP* - LISP for RT-11.
RAT* - RATFOR for RT-11.
MACLIB - Structured MACRO library.

Submitted by various authors on previous SIG tapes, and extracted by

Thomas J. Shinal
General Scientific Corp.
1681 East Gude Dr.
Rockville, MD 20850
(301) 340-2773

Several programming language systems for RT-11 have been put on previous Symposium SIG tapes. They have been collected into a "Languages" tape by Mr. Shinal. A few of the tape items are included here.

ALGOL, from the Spring, 1977 tape.
APL, from the Spring, 1983 tape.
FORTH, from the Fall, 1978 tape (?? hard to tell).
VLISP, from the DECUS library, version 1984.
RATFOR, from the Spring, 1983 tape.
MACLIB, from the Spring, 1983 tape.

The entire language tape will be provided to nodes of the RT-11 tape tree at a later date.

ALGOL1.DSK	6 Files,	49 Blocks
ALGOL2.DSK	21 Files,	436 Blocks
APL1.DSK	5 Files,	485 Blocks
APL2.DSK	13 Files,	304 Blocks
FORTH.DSK	9 Files,	291 Blocks
VLISP1.DSK	20 Files,	478 Blocks
VLISP2.DSK	44 Files,	482 Blocks
RAT1.DSK	11 Files,	471 Blocks
RAT2.DSK	31 Files,	457 Blocks
MACLIB.DSK	2 Files,	33 Blocks

****

MTPAPR - Description of MTLIB application.

N. A. Bourgeois, Jr.
NAB Software Services, Inc.
P. O. Box 20009
Albuquerque, NM 87154
(505) 298-2346

A description of an application of the MTLIB mag tape library (copyright NAB Software Services, Inc.). Data from a ROLLIN tape were recovered after ROLLIN failed.

MTPAPR.DSK	7 Files,	66 Blocks

****

The Fall, 1984, RT SIG tape contains 60 Files, 18375 Blocks.

It was prepared by:

R. W. Barnard
Sandia National Laboratories
Division 7523
P. O. Box 5800
Albuquerque, NM 87185

DCS - BARNARD

It is distributed by:

Thomas J. Shinal
General Scientific Corp.
1681 East Gude Dr.
Rockville MD 20850
(301) 340-2773

DCS - SHINAL

RT-32 OPINION POLL

At the last two DECUS U.S. Symposia there has been a growing interest in a high-performance, single-user, multi-job operating system for use on the MICRO/VAX family of computers. Dubbed RT-32, such an operating system would be expected to provide the same kind of performance, ease of use, versatility, and flexibility for the MICRO/VAX as RT-11 now does for the 16-bit world. In order to help DEC and DECUS determine the need for such a system, and what kind of features it should include, please fill in the following questionaire, and return it to

John M. Crowell
Crow4ell, Ltd.
145 Andanada
Los Alamos, NM 87544.

PLEASE pass copies of this survey to anyone else you think might be interested in this matter.

1. Do you have a need for an RT-32 operating system ?

   __ Yes
   __ No, but I need RSX-32
   __ No, MICRO/VMS fills my needs
   __ No, VAX/ELAN fills my needs
   __ No, RT-11 fills my needs

If you answered Question No. 1 no, you need not fill out the rest of this questionaire, but please return it with your response.

2. Describe briefly the type of environment in which you might use RT-32.

   ____
   ____
   ____

3. What features would you expect to see in RT-32, and what benefits would you expect to derive from RT-32 that are not provided by RT-11 ?

   a). ____________________________________ __
   b). ____________________________________ __
   c). ____________________________________ __
   d). ____________________________________ __
   e). ____________________________________ __
   f). ____________________________________ __
   g). ____________________________________ __

4. In the spcae to the right, please rank your answers to Question 3 inorder of their importance to you and your applications.

5. What features would you expect to see in RT-32, and what benefits would you expect to derive from RT-32 that are not provided by MICRO/VMS ?

a). ______________________________ ___
b). ______________________________ ___
c). ______________________________ ___
d). ______________________________ ___
e). ______________________________ ___
f). ______________________________ ___
g). ______________________________ ___

6. Please rank your answers to Question No. 5 as before.

7. What features and charactristics of RT-11 would you want to be included in RT-32 ?

a). ______________________________ ___
b). ______________________________ ___
c). ______________________________ ___
d). ______________________________ ___
e). ______________________________ ___
f). ______________________________ ___
g). ______________________________ ___

8. Please rank your answers to Question No. 7 as before.

9. What features and charactristics of MICRO/VMS would you want to be included in RT-32 ?

a). ______________________________ ___
b). ______________________________ ___
c). ______________________________ ___
d). ______________________________ ___
e). ______________________________ ___
f). ______________________________ ___
g). ______________________________ ___

10. Please rank your answers to Question No. 9 as before.

11. What features and charactristics of RT-11 would you want to be EXCLUDED from RT-32 ?

a). ______________________________ ___
b). ______________________________ ___
c). ______________________________ ___
d). ______________________________ ___
e). ______________________________ ___
f). ______________________________ ___
g). ______________________________ ___

12. Please rank your answers to Question No. 11 as before.

13. What features and charactristics of MICRO/VMS would you want to be EXCLUDED from RT32 ?

a). ____ __
b). ____ __
c). ____ __
d). ____ __
e). ____ __
f). ____ __
g). ____ __

14. Please rank your answers to Question No. 13 as before.

15. It might be possible to construct a single-user oprating system around a VAX/ELAN kernel. There would be some sacrifice of real-time performance and access to hardware. The resident kernel itself would probably occupy about 100kB. Would this type of operating system be suitable to your needs?

__ Yes, that would be OK.
__ No, that is unacceptable.
__ OK, but only as an interim mesure until a proper RT-32 could be devised.

16. Please include any additional comments and opinions you feel might be of benefit.

Name ______________________________
Address ____________________________
____
City,State __________________________
Zip Code _____________

Return Questionaire to:
John M. Crowell
Crow4ell, Ltd. *
145 Andanada
Los Alamos, NM 87544

* but not very

DECUS

DECUS SUBSCRIPTION SERVICE
DIGITAL EQUIPMENT COMPUTER SOCIETY
249 NORTHBORO ROAD, (BPO2)
MARLBORO, MA 01752

Bulk Rate
U.S. Postage
PAID
Permit No. 18
Leominster, MA
01453

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( ) Change of Address
( ) Delegate Replacement

DECUS Membership No.: ______

Name: ______

Company: ______

Address: ______

______

State/Country: ______

Zip/Postal Code: ______

Phone No.: ______

Mail to: DECUS - Attn: Subscription Service
249 Northboro Road, (BPO2)
Marlboro, MA 01752 USA

Affix mailing label