RSX-11/IAS SIG DECUS US CHAPTER

THE MULTI-TASKER

Volume 16, Number 3 September/October 1982

The Newsletter of the RSX-11/IAS Special Interest Group

TABLE OF CONTENTS

Columns

Articles

READ THIS FIRST

More on BRU Sorting

This issue has the third (and last) segment in the saga of BRU and its sorting algorithm. Carl has done a magnificent job detecting the problem and following it through. I would like to emphasize a couple of points:

1. The bug does not corrupt any tape that was written succesfully. If you do not get a sorting stack overflow error, the tape is OK.

2. Tapes must be restored with a version of BRU that has the same sorting algorithm as the tape was written with.

Disk Pack Crashes

The following note is from Donald Liston of Lawrence Livermore National Labortory. "Several disk packs have been seriously damaged at our facility due to a mechanical failure of a Digital RL01K-AA Disk Cartridge Cleaner. A metal clip holding the foam cleaning pad bent when a new pad was inserted, unknown to the person using the cleaner. When the cleaner was turned on, the clip scratched the surface of the disk as it cleaned!

The manufacturer of the cleaner, Inovatitive Computer products, has informed us that: (1) they are aware of the problem and have redesigned the faulty parts and are using in the current product, (2) these parts are the same as those used in the RK05-AA cleaner, (3) they cannot give use the serial number or date when the new parts began to be used.

It is HIGHLY recommended that all users of the RK01KK-AA or RK05K-AA Disk Cartridge Cleaner inspect the upper and lower arm assemblies in their units to determine if they have the old design. The old design can be determined by: (1) cranking out the arms manually (the crank is underneath the washer), (2) looking at the arms holding the foam cleaners; the tips of the clips should be visible. In the old design: (a) the tips of the clips are 1/8" wide rather than 1/16", (b) the tips are not sharply bent up into the housing, and (c) the arms of the clips are not tapered over the last 1/2 inch. Please note, if the assembly must be removed, use care not to lose the rubber seal between the upper arm assembly and the arm. You can contact Mike Davis of Inovation Computer Products at (213) 996-4911 for a more complete description of the old design. If you need new parts, order as follows:

UPPER ARM ASSEMBLY	Part #101156-1	$21.85
LOWER ARM ASSEMBLY	Part #101156-2	$15.60

From the Editor

This is it.

Charles Goodpasture of Transco Energy Company in Houston has accepted the job of Multi-Tasker editor, effective immediately. We are not sure when the next issue will be, either after the symposium or sooner if Charles receives enough submissions.

There is not much more to say. I have enjoyed being editor, especially because of the people it brought me into contact with. That part of the job I will miss. What I will not miss are deadlines, late nights, and sore fingers.

There are many people I need to thank, so many I am afraid to list them for fear of forgetting someone. Nothing makes the editor's job easier than high-quality submissions, so thank you to everyone who has sent me copy over the last year and a half. A special thanks to the DECUS publications staff, Martha Salinger, Donna Jacobs, and Chris Galipeau, for turning my rough mockup's into something you could read. And without the support of my family, I probably would have never got an issue to press.

Good-bye.

Ralph Stamerjohn
Former Multi-Tasker Editor
September 20, 1982

Chairman's Corner

As your new SIG chairmain, I thought it was about time that I brought you up to date on the SIG. There have been a number of changes of which you should be aware.

First of all, as most of you already know, George Hamma won the election for SUG Coordinator on the DECUS Board. George took office on July 1 and the SIG Executive Committee selected me to replace him as your SIG chairman. I would like to take this opportunity to express my tanks to George for the job he did as SIG chairman. I have just begun to realize the amount of time and effort George put it to make the SIG the success it has been.

There have also been two other resignations from the SIG Executive Committee. Phil Cannon has accepted a new job and is no longer involved with DEC systems. Phil was the Software Coordinator on the Executive Committee and has been primarily responsible for the success of the SIG Tape Copy program. Phil spent many all-night sessions at the symposia creating SIG Taps and he will be missed. Another member of the Executive Committee, Jim Neeland, has agreed to accept the position of Software Coordiantor. Jim has worked with Phil in the past on Tape Copy and is responsible for the SIG Tape from Atlanta.

Marg Knox has taken delivery of her VAX and is now busy trying to make it work. She will continue to be active in DECUS but in the VAX SIG. Marg was responsible for the excellent handouts which were available at the last two symposia. Marg has been an active and hard worker in the SIG and she too will be missed.

The following people have been appointed to the Executive Committee to fill the vacancies left by the resignations of George Hamma, Phil Cannon, and Marg Knox:

SYMPOSIA COORDINATOR	Jim Hopp
USER GROUPS COORDINATOR	Tom Viana
PUBLICATIONS COORDINATOR	Nancy Pallett

I would like to express my thanks to these people for being willing to accept this responsibility.

There are also new names on the Executive Committee from DEC and DECUS. Steve Paacola has a new job inside Digital, working on the POS operating system for the Professional 3XX computers. Laine Heiser is the new RSX Product Manager and will replace Steve as the Digital Representative for RSX on the Executive Committee. Tim Leisman is the new IAS Product Manager and will fill the Digital IAS Representative position on the Executive Committee. This position has been vacant for some time. The DECUS staff has also had a reorganizations and Debbie Kleiner has replaced Paula Morin as the DECUS Representative on the Executive Committee.

I warned you at the beginning that there had been a few changes. There are still a few positions open on the Steering Committee. If you are interested, please contact me and I will try to explain what is involved.

ENOUGH OF THAT!

The 1982 Menu and Ballot were distributed in the August issue of the Multi-Tasker. The Menu is the most effective was the SIG has of influencing the way DEC expends its resources in developing the products we use. The more responses which are received, the more effective it will be. If you have not returned your ballot, please take the time to fill it out and send it in. A new format for the Menu and ballot is being tried this year. If you have comments, criticism, or suggestions on the format, please let me or Louis Stoll know.

The time for the Fall Symposium is rapidly approaching. The RSX/IAS SIG has 57 sessions scheduled and the are over 100 related sessions scheduled by other SIG's. The first article in this issue has much more information.

On Friday at 3:30 (I know it is late), there will be a SIG planning session. This will be the planning session for St. Louis Symposium. If at all possible, make your travel arrangements so you can attend this session. The only way we have of knowing what type of symposia programs you would like is for you to come to this session and tell us what you like and don't like.

Legare Coleman
RSX/IAS SIG Chairman

From Five Years Ago

Gail Green
Multi-Tasker Historian

The featured installation in the September 1977 issue (Vol. 8, No. 3) of The Multi-Tasker was ASARCO of South Plainfield, New Jersey. Applications on ASARCO's RSX-11M system included data collection for an electron microprobe and a database system for logging chemical samples.

The reinstated SPR column included:

- A report of problems in RSX-11D V6.2 with VT05 direct cursor addressing working differently than it did with RSX-11D 6A.

- The RSX-11M function IO.RNE echoed a carriage return/line-feed if there was no output to the terminal before the next request for input. This caused two problems with switched packet networks: (1) increased communication costs and (2) the CR/LF could interrupt in the middle of a line of input.

The June Multi-Tasker contained a questionnaire concerning Multi-Tasker readership, the usefulness of DECUSSCOPE, and user reaction to the possibility of Multi-Tasker subscription fees. Sixty percent of the respondents voted for the elimination of DECUSSCOPE. However, the results were not analyzed, just published in tabular form, to be passed on to various DECUS executive boards. Many of the comments received with the completed questionnaires were also published. The comments were varied, but the common thought that came through was the usefulness of The Multi-Tasker to its users and the cry for its continuing existence.

The October 1977 issue of the Multi-Tasker was a big issue. In addition to the regular columns - "Chapter Happenings", "Installations", "Suggestions", and "SPRs", there were articles featuring:

- James Downward's description of improvements for the accounting facilities for RSX-11M V3 and of general problems with V3. This was the start of the KMS Fusion kit!!!

- David Stern's description of University of Colorado's software for monitoring terminal connect time and memory and disk usge.

- More words from James Downward concerning DEC software that didn't work and DEC's response (or lack of) to those problems.

- Confirmation of problems reported previously with 11D V6.2.

A new column, "Hardware Hints and Kinks Mini-Newsletter", appeared in this issue. The purpose of this new column was "to gather, attempt to verify, and disseminate useful information on hardware fixes and improvements in a timely manner", an SPR-type column for hardware. Excerpts from this column follow. Some of the information may be dated, but there are a lot of older model PDP-11's out there still up and running.

- UNIBUS PROBLEMS

  Yellowed and delaminating UNIBUS cables. Old Hughes UNIBUS cables turn yellow, then green, and seem to delaminate. In all known cases this has been a non-problem. Delamination is apparently only the laquer coating and coloration is not due to copper wires coming throug.

- STATIC ELECTRICITY AND NOISE PICKUP

  * Get rid of pantyhose, carpeting, etc.

  * Static tends to zap the interrupt enable bits on termninal interfaces. If your DL-11 interfaced terminal suddenly refuses to communicate, set the interrupt enable bit in status register using console switches or another terminal and OPE.

  * Ground everything in sight and assure front panel grounded.

  * Australia Wool Board says 80 to 100% wool carpets help. We have not verified this.

  * EMI interference sometimes acts like static. Beware of intermittent problems which correlate with startup of motors, passing of railroad trains, etc.

  * On 11/70 CPU's placing key in lock position leaves some gates floating, and susceptible to noise pickup.

- DL11 PROBLEMS

  Crosstalk. DL11's adjacent to each other or certain other devices may experience crosstalk, causing garbled characters. Separate them by blank SPC slots or by insenitive devices such as the bootstrap module.

- RK05 DISK DRIVES AND CARTRIDGES

  * There is an "absolute filter" inside each RK05 drive (not to be confused with the foam pre-filter visible at the back) which must be replaced periodically to prevent head creahes. If you leave power on continuously, this filter is good for about 4 months.

  * RK05 Cleanliness Proof Test. Editor's note: Users interested in a proof test for evaluating the effectiveness of RK05 disk and head cleaning may wish to refer back to this issue.

DECUS/RSX SIG Library News

Paul Tompkins
Library News Editor

Over the years, DECUS, through the DECUS library, and the RSX-11/IAS SIG, through the SIG tapes, have accumulated a huge set of useful software. If you have news about any of this software, please send to the Multi-Tasker c/o this column. This includes any problems discovered, patches to existing software, short notes on library submissions you found useful, or any other information you may have. Send submissions to Multi-Tasker - Library News, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 10752.

Most Frequently Ordered DECUS Library Programs

During May 1982, the program/tapes most frequently shipped by the DECUS Library, on a world-wide basis, were as follows. The list includes the top sixteen programs (or tapes) shipped during the month. Numbers in parentheses on the right margin indicate the quantities shipped.

1)	11-SP-12	RSX Special Collection	(20)
2)	11-SP-18	C Language System	(18)
3)	11-435	FORTRAN Graphics Support for the VT105 Version: Apr'80	(17)
4)	11-SP-10	RSX Special Collection, created March 1980	(17)
5)	R11-LIB-1	RSTS-11 Library Tape #1	(13)
6)	11-346	PASCAL Compiler, Version: October 1981	(12)
7)	11-SP-11	RT-11 Special Collection	(12)
8)	11-451	RT-11/FORTRAN Support for Data Aquisition and Display on a VT105, version: 26 August 1980	(11)
9)	11-456	DUPLEX: Serial Communication Between Computers	(11)
10)	R11-LIB-2	RSTS-11 Library Tape #2	(11)
11)	11-370	DUNGEON, Version: 2.5, October 1980	(10)
12)	11-417	MINC BASIC/FORTRAN IV, Virtual Terminal Support	(10)
13)	11-232	FORTH: Programming System for the PDP-11, Oct'78	(9)
14)	11-412	MINC/FORTRAN Support for MNCKW, VT100, DLV11J	(9)
15)	11-SP-16	Symposium Tape from the PASCAL SIG, Spring 1980	(9)
16)	11-314	RT-11 RUNOFF, Version: M01-C, April 1980	(8)

New Submissions to DECUS Library

The following new/revised programs are not listed in the 1982/1983 Catalogs.

11-SP-6	DDT22: Mapping DDT/Sysaid Package (Revision) 600' Magtape (MA)	RSX-11D,M,S,IAS
11-SP-23	Spring 82 Canadian RSX Symposium Tape	RSX-11M V3.1/3.2

	(Revision) 2400' Magtape (PA)	
11-507	A Program for Saving Deleted Files (Revision) Listing (BA)	RSX-11D,M
11-527	RSX to RSX Communications Utility (XFR/XSL) (New) 600' Magtape (MA), Floppy Disk (KA)	RSX-11M V3.2
11-538	PLOT-55 for RSX-11 (New) 600' Magtape (MA)DOS-11, Floppy Disk (KA)RT-11	RSX-11M V4.0, M+ V2.0

RSX-11M V4.0 Software Tool Fixes

Joe Sventek

Lawrence Berkeley Laboratory
Berkeley, California 94720

1. Version 4.0 Software Tool Fixes

The following two correction files, applied to files distributed in [307,31], are required to permit background processing on systems with multiple CLI support. The should be applied to the copies of the files in ~bin/, and then BSPBLD.CMD should be executed.

***** bspawn.cor *****

```
bspawn.src;2/au:72.=bspawn.src;1
-156,,/;V4/
        .if df  a$$cli
clinam: .rad50  /MCR   /
        .endc
-337,,/;V4/
        .if df  a$$cli
        mov     r1,r0                   ; need UCB address for $stcli
        clr     u.cli(r0)               ; clear all bits
        mov     #clinam,r3              ; address of CLI name
        swstk$  35$                     ; switch to system state
        call    $stcli                  ;# set CLI to MCR
        return                          ;# return to user state
35$:
        bcs     40$                     ; c set => error in setting CLI
        mov     ucb,r1                  ; restore registers
        mov     offset,r2               ; ...
        .endc
/
```

***** bttab.cor *****

```
bttab.src;2/au:72.=bttab.src;1
-32,35,/;V4/
        .if df  a$$cli
        .word   0                                   ; CLI pointer
        .endc
        .if df  m$$mup!a$$cli
/
```

Some of you may have noticed that 2 or 3 of the task images generated during toolgen under version 4.0 linked to the PLAS FCSRES overflow the 32KW task image limit. (Actually, they don't at task-build time; the problem manifests itself when the tasks are run and cannot extend themselves for the required I/O buffers, babbling

```
TSKNAM *** error extending task
```

and doing no real work.) A fix is due in a maintenance release at the fall symposium. In the mean time, a workaround can be achieved as follows:

1. Move to a convenient working directory
2. Install the shell as

   ```
   INS ST:[105,1]sh/task=...shx
   ```

   This is necessary since the history shell, which is usually installed as "...SHL" is one of the images with the problem we are trying to fix.
3. Copy ~bin/tools.tkb to the current directory. Edit this copy to delete the LIBR=FCSRES:RO line.
4. Invoke shx; using the tool `ld', relink the offending tasks with command lines of the form

   ```
   ld -v -phsh ~bin/hsh
   ```

   The -p flag causes the task image hsh.tsk to be created in the current directory. Repeat this process for each tool which has problems.
5. Test out the new utilities to make sure that the problem is gone. If they work correctly, copy the task images to the ~bin/ directory and re-install the history shell.

The reason that this procedure works is that most tools which have to use a known file (such as ld using tools.tkb) look for the file along the standard search path (current, home, ~usr, ~bin). By making a special copy of tools.tkb in the current directory, ld will find that one instead of the one referencing FCSRES in ~bin. If you have not used FCSRES in taskbuilding the tools, you should not have experienced this problem.

2. Home Directory Management Corrections

The following are the required corrections necessary to provide the home directory capabilities which I described in an article `in a previous issue of the multitasker (if I could find my copy I could give you the volume and issue number). As many of you probably guessed, those corrections were for version 3.2. The necessary correction files for version 4.0 follow. I have been using them since June with no apparent problems.


```
***** drgtk.cor *****

OU:DRGTK.MAC;2/AU:72./-BF=IN:[11,10]DRGTK.MAC;1
\
-27
;
;       JOSEPH S. SVENTEK, 28-APR-82
;
;               JS001 -- RETURN U.LUIC IN PROTECTION WORD IF M$$MUP
%
-35,,/; JS001/
        .NCALL  UCBDF$          ; DEFINE UCB OFFSETS
        UCBDF$

-135,,/; JS001/
        .IF DF  M$$MUP

        MOV     T.UCB(R5),R0            ; GET TI: UCB ADDRESS
        MOV     U.LUIC(R0),(R3)         ; COPY LOGIN UIC

        .IFF
-137,,/; JS001/

        .ENDC

/


***** hello.cor *****

OU:HELLO.MAC;2/-BF/AU:72.=IN:[12,10]HELLO.MAC;1
\
-98
;
;       JOSEPH S. SVENTEK, 28-APR-82
;               JS001 -- ASN SY:=HO:/LOGIN
%
-124,,/; JS001/
HOME:   .ASCII  %ASN SY:=HO:/LOGIN%
HOMLEN=.-HOME
-887,887,/; JS001/
170$:                                   ; Reference label
        MOV     #HOME,SPAWN+S.PWCA      ; BUILD DPB FOR HO: ASN
        MOV     #HOMLEN,SPAWN+S.PWCL    ; ...
        DIR$    #SPAWN                  ; SPAWN ASN COMMAND
        BCS     171$                    ; C SET => ERROR
        STSE$S  #EFN2                   ; WAIT FOR IT TO COMPLETE
        BCS     171$                    ; REAL BAD KARMA
        CMP     EXST,#EX$SUC            ; SUCCESSFUL COMPLETION?
        BEQ     172$                    ; YES
171$:
        MOV     #ER13,R0                ; Spawn failure
        CALL    $EROUT                  ; Notify user
172$:
        MOV     #ODPB,R4                ; GET TERMINAL DPB ADDRESS
/
```

Of course, I couldn't resist adding quite a few more "pseudo-DCL" commands to TDX such as FREe, SPOol, PURge, COPy etc. It's both well documented and easy (just use the supplied commands as models). Note that because TDX makes use of the immediate exit feature of the RPOI directive, TDX can be used recursively. I have a SORt command in my version of TDX which generates a SRD command line. But since SRD is not installed, TDX is called once more to do a flying install (its not as fast as lightning, but it is easy and it works).

It's in the Code

Jim Preciado
Column Editor

Welcome to the first of a hopefully continuing Multi-tasker feature. The purpose of "It's in the Code" is to serve as a collection point for all known and newly discovered undocumented RSX-11M/M+ features. All versions of RSX-11M and M+ are eligible. Carry overs from IAS are also fair game. My aim is to collate information on features and functions that are hidden in these operating systems. Submissions will be sorted based on operating system and version and will appear monthly in this column. After a sufficient number of submissions have been received, they will be gathered into an index and this index will be published as a reference tool. I also hope to highlight past Multitasker articles dealing with such features.

Now for the ground rules. I will not personally be looking for these features. I could not possibly find them all and I do need to make a living. Also, I'm sure that users already have found all sorts of things hidden in the executive and utilities. So, get busy and send in these tips. Submissions should deal with features that already exist in the executive. New functionality that involves adding code to the exec or a utility might better dealt with seperately. If a hook for a feature already exists and we just have to add a missing piece, the submission will be considered. If a conditional assembly or a branch statement must be removed to enable a feature, let us know. If something is there as is and ready to be used by everybody, terrific - tell us how it works! Submissions should be sent to The Multi-tasker - It's In the Code, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752. All submissions published will contain the name of the person sending it in.

Here are some hints for finding submissions.

1. Read the source code (that's how this column got its name). The RSX distribution kits contain the sources to the entire executive as well as things like the LoaDeR, MCR, HELlo, and IND (ICP for version 4.0). If you can spare the space during SYSGEN, save and print the listing files. They show not only what was assembled but, perhaps more important, what code was bypassed due to conditional assemblies. Those of you with RSX source licenses can help us less affluent users by looking at utility and ACP sources.

2. Carefully study the existing documentation. Some features may get only a one line mention and become undocumented just by becoming lost in the pile. Also, some oddities may be documented and could fall in the "It's not a bug, its a ..." catagory. I will try to include SPRs dealing with things like this in this column.

3. Look at the documentation for earlier releases of the operating system. Just because a command or switch was dropped from the documentation in one release to another doesn't mean that it was dropped from the code.

4. Try it! It might work! An undocumented feature may also be the product of a careful manipulation of existing abilities. However, these sort of submissions might better be dealt with as an entry in other columns ("The Wizard's Book of Magic", for example) or as a full feature themselves. If in doubt, send it in.

Lastly, here's a goody to start off the column. It was given to be me a fellow user who came to my site from an RSX-11M V3.1 system. Get those cards and letters in folks!

Submission Number: 1.0

Operating System: RSX-11M V3.2
Component: PIP
Feature: /RM switch

Submitted by: Steve Legensky
Advanced Technology Systems

Format: [g,u]filename.ext;ver/RM

This switch removes a file from a directory. It removes only the pointer to the file-id. File-ids are contained in [0,0]INDEXF.SYS. PIP /RM does not go out to INDEXF.SYS and check the validity of the file header. This makes it possible to delete a file whose name exists in a directory but does not point to itself in INDEXF.SYS. This switch is useful for removing dangling directory pointers. In my case, I was able to remove a directory file from [0,0] after a user had deleted the directory by file-id. I could not delete the file with PIP because the file header was marked free and I could not create the directory again using UFD because an entry with the same filename already existed in [0,0]. If in accessing a file, you receive the messages

FILE ID - SEQUENCE NUMBER CHECK or

NO SUCH FILE

it could be because the directory entry and FILE ID have gotten seperated. PIP /RM can be used to correct this situation.

From The Wizard's Book Of Magic

The Magic sessions at the symposium have become one of the most popular features of the RSX/IAS SIG. This column has the same purpose: to exchange and discuss ideas on non-standard RSX and IAS programming. Readers are encouraged to submit items to this column and are also warned that the material here have not been checked for accuracy. Also, implementation of any items from this column will be completely unsupported. The material here is potentally dangerous: incorrect usage could result in system crashes and other incorrect system operations. Send any submissions to Multi-Tasker - Magic, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 10752.

Adjusting the System Memory Size for RMDEMO

Micheal E. Mazzoni

Process Control Systems
1300 S. Calhoun Road
Brookfield, Wisconsin 53005

RMDEMO automatically adjusts its resolution to system memory size. That is, the same horizontal space is used for a 64 KW system as a 256 KW system. The KW's per inch of display is increased as the system memory size is increased.

When the CPU is a 18-bit PDP-11 with a full set of memory (128 KW), only the first 124 KW is displayed, because the I/O page occupies the top 4KW of address space. This results in:

- An uneven distribution of memory increments, e.g. 0-7-15-23-31.
- No display of I/O page device commons.

Fortunately, there is an extremely simple fix. RMDEMO looks at the value of the global symbol $SYSIZ every time it updates the display. Modifying the value $SYSIZ with the OPEn command will produce the desired result. For example:

```
OPE 7742                ! location of $SYSIZ on MY system
007742/ 007600 10000    ! old: memory stops at  7600(00)
                        ! new: memory stope at 10000(00)
```

The location of $SYSIZ is found in the executive map - RSX11M.MAP. Note, that value must be changed with an OPEn after the system is booted because SAVe will size memory and set the initial value.

With the value $SYSIZ at 10000(8), RMDEMO will sow 128 KW for system size. Device commons will be displayed and there will be a nice even distribution of memory increments.

Autobaud Detection for Gandalf's

Robert J. Kobrin

Mobil Research and Development
Paulsboro, New Jersey 08066

The RSX-11M V4.0 full-duplex terminal driver fails to autobaud lines from a Gandalf Private Automatic Computer Exchange. This occurs because the Gandalf system raise Ring Indicator (RI) and Carrier Detect (CD) simultaneously. On detecting RI, module TTMOD starts a timed wait for CD. Since the DH/DM-11B interrupts on change of state and CD is already high, no interrupt occurs. TTMOD eventually times out, dropping the line.

The following patch allows Gandalf lines to autobaud correctly at speeds up to 4800 baud.

```
TTMOD.MAC;2/-BF/AU=TTMOD.MAC;1
\
-2,2
        .IDENT  /V3.00K/
-41
;       RJK01   15-APR-82       R.J. Kobrin, Mobil R&D, Paulsboro NJ
;               Answer call when RI and CD/CTS are simulatenous
-63
CTS     = 40            ;Clear-to-send bit in DM11 line reg
CD      = 100           ;Cariier-detect bit in DM11 line reg
-88
;       2.1.1   If CD is absent, wait for it (or time out)
;       2.1.2   If CD is already on, go to 3
%
-147,147,/;RJK01/
        BIT     #CD!CTA,@R3     ;;;Are CD and CTS already on?
        BEQ     20$             ;;;No - start timed wait for them
        BR      MANS            ;;;Yes - answer right now
/
```

Fall 1982 Symposium Preview

Ralph Stamerjohn
Multi-Tasker Editor

The Fall 1982 DECUS Symposium will be held at the Disneyland Hotel from December 6-10, 1982. While I once thought Atlanta would be impossible to top, the idea of the Magic Kingdom and the RSX SIG getting together should do the trick.

The RSX-11/IAS SIG has a new symposia coordinator, Jim Hopp of Swift Research and Development. Jim is brand new to the job but his first effort does not show it. The following is a synposis of the planned sessions:

- RSX/IAS BUSINESS SESSIONS

  * RSX-11/IAS SIG ROADMAP AND BUZZWORDS

    The Roadmap and Buzzwords session opens the week's activities for the RSX-11/IAS SIG. We will highlight sessions sponsored by the SIG and announce any last minute changes. Special note will be made of the various places to get your questions answered: the campground, software clinic, question and answer sessions, product panels, or SIG business meetings.

    This is a long and exiting week, so the second half of the session will point out various survival techniques. We also have collected over 100 RSX/IAS buzzwords. The more obscure of these will be defined so all attendees, both new and old, speak the same language for the rest of the week.

  * RSX-11/IAS SIG OPENING SESSION

    This session is the start of the SIG business for the week. This will be the opportunity for the SIG members to be updated on the status of the various activities of the SIG. Agenda items include: introduction of the SIG Steering Committee, introduction of the Digital representatives, reports on current SIG activities, and a policy question and answer session. All RSX/IAS SIG members should attend this meeting.

  * RSX-11/IAS SIG WORKING GROUPS SESSION

    This will be an informal session during which all of the existing working groups sponsored by the RSX/IAS SIG will meet in parallel. The meeting will start with brief reports from the working groups. Then each group will move to a part of the room and conduct their own meeting. Anyone is welcome to join a working group or form a new one. Note, some of the major working groups have setup additional meetings for further business (see below). But all working groups will have meetings at this session.

  * IAS WORKING GROUP SESSION

    The IAS Working Group is developing many projects, some of which include: performance and features versus other operating systems, conversion from IAS to other systems, IAS programs from past SIG tapes, conversion of new utilities to IAS, and feedback to Digital on the future needs of IAS users. If any of these projects are of interest to you or if you have ideas for other projects, please come to this interactive session.

  * RSX SIG TAPE WORKING GROUP SESSION

    The RSX Sig Tape Working Group will meet to discuss methods for improving the speed of distribution of the Fall 1982 SIG tape and review and combine information collected from various sources for generation of a "Best-of-SIG-tapes" collection. We will also discuss arranging for resubmission of the old FLX SIG tapes to the DECUS library in BRU format.

  * UNSUPPORTED VERSIONS OF RSX-11M WORKING GROUP SESSION

    This session is intended to benefit those users of RSX-11 who, due to small system size, lack of funds, etc. decide not to upgrade to current supported versions of RSX-11M. The working group is designed to provide user-to-user support mechanism for patches and adding some features of newer versions, support for new devices on older systems, and experiences with DECUS programs. All users of RSX-11M versions 3.1 and earlier and those users not converting from V3.2 to V4.0 are welcomed.

  * RSX SIG LIBRARY EVALUATION WORKING GROUP MEETING

    The DECUS Library Evaluation Working group has been looking at 37 packages from the RSX and VAX areas of the DECUS library. This working group meeting will be directed toward collecting evaluation forms, answering any questions, fixing tree distribution problems, and encouraging new members to join and evalutate at least one package in exchange for the total collection. Evaluation forms will eventually be submitted to the DECUS staff for use in catalogues, archiving decisions, and other such purposes.

  * RSX/IAS MENU GENERAL MEETING

    The RSX/IAS SIG maintains a "menu" of issues and product enhancements which are submitted by the SIG membership. An annual poll is taken (last issue) to determine which of the items are most important to the user community. The results of this poll have been an important factor in determining how Digital allocates its resources in the product development cycle.

    The 1982 ballot voting will be complete and tabulated by the time of the symposium. The results will be presented to the membership and Digital will make a response.

* THE RSX-11/IAS SIG IN REVIEW

The RSX-11/IAS SIG is in a constant state of change. We need to think about who we are and how we can best serve ourselves. This session is an open-floor discussion by the attendees on what the SIG is doing right, what activities are outdated, and what new ventures we should explore. This is your chance to answer some key questions facing the SIG and open discussion on new problem areas.

* RSX-11/IAS SIG CLOSING SESSION

The symposium is not over until after you attend the SIG closing session. All SIG members are urged to attend this session and let the SIG leadership and Digital representatives know your opinion of the week's activities. All significant events of the week will be reported. Your inputs are important as they will be used for both future Symposia and SIG planning. Also, at the end, a final question and answer session will he held with both SIG Steering Committee members and Digital representatives available.

* RSX/IAS SIG PLANNING SESSION

This session will be a round table discussion and planning meeting for the St. Louis symposium. The content and format of traditional symposia sessions will be discussed and ideas for new sessions for St. Louis solicited. Everyone is invited. This is your opportunity to get that session you have always wanted to attend scheduled for St. Louis.

o QUESTION AND ANSWER SESSIONS

* RSX-11M AND RSX-11M-PLUS NEW USERS QUESTION AND ANSWER SESSION

At this session, a panel of experienced RSX-11M and RSX-11M-Plus users will answer questions posed by less experienced users. typical to new sites and new users. Most attendees at the Symposium fall into this class. This session is for you. Questions too technical or which fall outside the expertise of the panel will be directed to the main RSX-11M Question and Answer session.

* RSX-11M AND RSX-11M-PLUS QUESTION AND ANSWER SESSION

During this session, a panel of Digital developers will attempt to answer all technical questions related to RSX-11M, RSX-11M-PLUS, and layered products. This session provids a public forum for technical and policy interchange with Digital personnel.

* IAS QUESTION AND ANSWER SESSION

This session is the high point of the symposium for the serious IAS user. Several IAS experts and a room full of experienced users respond to questions from the floor. Bring your IAS problems, questions, and experiences to this valuable session.

* QUEUE MANAGER QUESTION AND ANSWER SESSION FOR RSX-11M

This session is aimed at users with an interest in a more detailed understanding of the usage and operation of the RSX-11M queue manager system. Technical questions related to using the queue manager along with those relating to interfacing to it are requested.

* RSX SOFTWARE CLINIC

The RSX/IAS Software Clinic is the place for a programmer to bring her/his specific problems. As you walk in, a triage team will assess your problem and assign you to the "doctor" best able to answer your question. The clinic's doctors are experienced RSX/IAS applications programmers, system experts, and Digital developers. The clinic is for all users, but especially first-time DECUS attendees and new RSX/IAS users. For specific problems, please bring sufficient documentation (i.e. program listings, error messages, crash dumps). Also if you do not have problems but would like to be a doctor, feel free to drop by and join in.

o DIGITAL SPONSORED SESSIONS

* RSX-11 PRODUCT PANEL

In this session, Digital Product Management will present the status of the RSX-11 family and provide some information on the future of RSX-11. This is a non-technical session and provides a forum to get answers to policy questions.

* IAS PRODUCT PANEL

Digital Product Management will conduct a panel on IAS policy. This will be followed by a question and answer session for questions related to policy issues and other areas of general interest.

* RMS PRODUCT PANEL

This session is an overview of RMS Version 2. Included in the discussion are new access methods, new functionality in standard utilities, and new utilities.

* RSX: A FAMILY PORTRAIT

Like all families, RSX-11 has many stellar members and a few skeletons we keep locked in the closet. This session will give you a little bit of the family history and may even include a few words about the skeletons, from the days of RSX-11A, B, and C to the current releases. The presentation will be more historical than technical, but it will answer many of your questions about why we did what we did and when we did it. You may never make fun of FLX again.

* WHY DO CARE ABOUT RSX-11M-PLUS?

RSX-11M-PLUS is the latest in the series of RSX-11 operating systems, an evolution of RSX-11M. RSX-11M-PLUS is not the solution for everyone, but it is the RSX-11 system being emphasized by Digital. At a functional level, the differences between RSX-11M V4.0 and RSX-11M-PLUS V2.0 will be presented in this session. Digital will also provide some information on where the RSX-11 family is going.

* P/OS - AN OFFSPRING OF RSX-11M-PLUS

This session will describe the modifications that were made to RSX-11M-PLUS that resulted in the new P/OS system. The new functionality added and some old foundations that were deleted will be described, along with some unexpected surprises found along the way.

* PDP-11 OPERATING SYSTEM COMPARISION FOR RSX USERS

Now that all PDP-11 operating systems can run on the smallest systems, it is increasingly important to understand the similarities and differences between the various systems to choose the optimal solution. This session will give an overview of RT-11, RSTS/E, DSM-11, and MicroPower/PASCAL for the user familar with RSX-11 systems.

* 16-BIT SOFTWARE PRODUCT SERVICES STATUS AND MENU

This session is to provide Digital with the opportunity to report on the status of Software Product Services for the 16-bit user. Digital managers will respond to questions and comments from the floor. Menu items will be solicited for response by Digital at the next symposium.

* HOW TO CUSTOMIZE THE RSX-11 COMMAND INTERFACE USING DCL AND TDX

This talk will describe two methods of adding custom user commands to RSX-11M and RSX-11M-PLUS. The first ten minutes will describe the system catchall task, TDX. TDX provides a simple way for users to add their own commands to MCR or DCL or to define a default action for the system to take on receiving unrecognized commands. The rest of the talk will describe how to use DCL's Macro Meta-Language (MML) to add new commands to DCL or modify existing

DCL commands. MML is a specialized syntax description language based on BNF and is implemented as Macro-11 macros. It is used to define all RSX-11 DCL syntax and its equivalent MCR syntax.

* MIGRATING RSX-11M DRIVERS AND PRIVILEGED TASKS TO RSX-11M-PLUS

Modifications to RSX-11M drivers and data base will be discussed, illustrating the relative ease of conversion. A discussion of the required conversions to migrate a privileged RSX-11M-PLUS task will cover mapping considerations, the GIN$ directive, and data structure differences including implications of executive headers and secondary pool TCB's.

* USER WRITTEN CLI'S UNDER RSX

This session will offer insights and information on how and why to write a CLI under RSX-11M/RSX-11M-PLUS. Topics covered will include: advantages of a user-written CLI, the executive directives that support CLI's, and putting it all together into a CLI.

* RSX DEVELOPMENT WITH VMS

The RSX development team is now responsible for AME, the Applications Migration Executive portion of VMS. This session will compare the AME with RSX-11M V4.0 and the development efforts underway to improve the compatibility.

* HOW TO BUILD APPLICATIONS WITH CLUSTER LIBRARIES

The purpose of this session is to provide an overview of the concept of "clustered" memory resident libraries, as well as a discussion of the enhancement of virtual address space that clustering provides for most high-level language applications. The "mechanics" of how to build an application task that uses clustered libraries will be presented, including examples using RMS-11, FMS, and language OTS systems.

* ACP'S IN IAS WITH EXAMPLES FROM F11ACP

This session will desribe the function of Ancillary Control Processors (ACP's) in the IAS operating system. Services provided by the ACP will be discussed as well as the interface between the executive, device handlers, and the ACP. Data structures maintained by the executive on behalf of ACP's will also be included. Digital's F11ACP will be used as an example of the role the ACP in the system.

* POSSIBLE SOLUTIONS TO THE SCOM NODE PROBLEMS ON IAS

Many IAS systems have at one time or another come to halt due to the lacke of free nodes in the system common area (SCOM). Part one of this session will identify the users of SCOM nodes and discuss various methods of reducing an application's node requirements. The second part will focus on various mechanisms of increasing the

number of nodes available in SCOM. These include removing various routines currently included in SCOM and creating a special partition used to supply nodes for some data now held in SCOM.

o PANELS, WORKSHOPS, MAGIC

* SITE USE OF RSX SIG TAPES

Several knowledgable RSX users will discuss their own site's usage of various items they have culled from the massive collection of the RSX SIG tapes. For this symposium, rather than just a listing of useful programs, the panel will attempt to indicate in greater detail just what the programs do and how they have made use of them.

* IAS MAGIC

This session is a forum for all IAS users to learn of programs - both new and old - and compare experiences and ideas with users from various hardware and application backgrounds. Several programs and techniques will be presented, followed by an open discussion driven by topics from the floor.

* THE LORE OF RSX-11M

The "M" in RSX-11M does not stand for Mature, or for Moldy; its stands for Mellow. Great civilazations of the past shared their knowledge by passing on its legends and wisdom by word of mouth. Although RSX-11M is by no means an ancient and forgotten art, an amazingly large body of knowledge exists among its users, often passed on in the form of War Stories and other less-than-formal means. The purpose of this session is to provide a forum for all those who wish to tell their story, pass on their discovery of some system quirk, or crow about their new and unique solution to a problem. Wizards, apprentices, and minions are all invited.

* RSX-11M SYSTEM PROGRAMMING SHORT NOTES

People are always discovering new ways to invalidate their RSX-11M software warrantry by modifying RSX-11M. However, modifications also open the doors to solving real-world problems. Anyone with RSX-11M "tricks-of-the-trade" is invited to speak briefly about them at this session. In order to get some degree of organization, speakers are asked to write up a breif abstract before the session and drop off in the RSX Campground. New topics will also be taken at the session. There is also a place at this session for looking into problems no one has been able to solve yet, commonly called "blueskying". It is expected that this session will get deep into the internals of RSX-11M, with buzzwords and bits flying left and right.

What about RSX-11M Magic?

As you may have noticed, there is no RSX-11M Magic session. RSX is leading the way again. Being the first to have Magic sessions, we are also the first to kill them off. Magic no longer worked. The floor-show aspect and large rooms made the original intent, free-flying discussions on internals, impossible. The previous two sessions fill the gap. "Lore" is the place for the fun and games. "Short Notes" will be serious.

o USER TUTORIALS AND PAPERS

* SUPERMAC TUTORIAL

SUPERMAC is a set of Structured Programming Macros for MACRO-11 which enables the programmer to write efficient Assembly Language code while using control constructs otherwise available only in high-level languages. SUPERMAC contains IF-THEN-ELSE, WHILE, CASE and REPEAT-UNTIL constructs, as well as many other language features which enable the programmer to write readable, block-structured programs. This tutorial will provide an introduction to SUPERMAC, with examples of its use in writing both system and applications software.

* THE DATA ACQUISITION, DATA REDUCTION AND CONTROL SYSTEM (DARCS) FOR THE NRCC 2x3m WIND TUNNEL

In 1978, a new data acquisition, data reduction and control system was designed for the NRCC 2x3m wind tunnel. Since that time, the system has ben installed and has evolved through three major hardware and software development stages. The purpose of this talk is to outline the current system and to provide the background to its development. Generally, the result is that the utilization of this test facility has increased by a factor of five. In addition, data can now be displayed graphically in real-time. Repetitive test sequences can be automated in order to reduce the amount of test-time required. Thus, the effects of wind-produced forces and motions on scale models of elastic structures can be studied in a controlled test environment. Finally, the effectiveness of proposed solutions to structural problems which reveal themselves during the test can be investigated quickly.

* THE MATHEMATICS OF RSX-11M

RSX-11M can be viewed as a finite-state system, the behavior of which is governed by a set of operating principles. Understanding of these fundamental principles can ease the solution of application problems, particularly those involving the interaction of several tasks. This session will walk through the basic axioms and thereby attempt to build a better understanding of why RSX-11M (mis)behaves the way it does.

* A PDP-11/34 BASED REMOTE TERMINAL EMULATION SYSTEM

Remote terminal emulation is a form of computer performance evaluation that uses an external driver to simulate the workload that is normally imposed on an interactive computer system. This system, called RTE11, is based on a PDP-11/34 running under RSX-11M V4.0 with DH11 equivalent asynchronous multiplexers providing the terminal interface capability for up to 64 lines.

* SRD -- THE DECUS SORTED DIRECTORY UTILITY PROGRAM

SRD is a user-written RSX-11 utility that sorts a user file directory, then outputs it to a file or to the user's terminal. SRD can also select files by date and by parts of the file specification. This session is intended for users unfamiliar with the DECUS Library or the RSX SIG tapes who wants to learn how this program can be useful in his or her installation.

* MIGRATION FROM M TO M-PLUS

During the last year The Record has chosen to migrate from RSX-11M Version 3.2 to RSX-11M-PLUS Version 2.0 (with a brief stop at Version 1.0). This session will discuss our reasons for that choice, and will discuss our experience of the migration.

* SHADOWING VIRTUAL DISKS UNDER M-PLUS

We have combined the Decus-developed Virtual Disk package with the ability of M-Plus to "shadow" (redundantly record) a disk. This allows us to selectively shadow only certain critical files instead of duplicating entire three hundred megabyte disks. We shadow files from several physical disks on a single, physical "shadow" disk. This session discusses the technique used, advantages gained, limitations, and problems (particularly operational) that were encountered.

* AN RSX-11M DEVICE DRIVER IMPLEMENTING A NETWORK PROTOCOL FOR THE DR11W

At Fermilab, DR11W's have been used as high speed data links to interconnect PDP-11's (under both RT-11 and RSX-11M) and VAX's (under VMS), in data acquisition applications. Using this hardware, several processors can be interconnected to provide distributed data collection, data monitoring and control for physics experiments. This paper discusses a device driver implemented under RSX-11M. The driver allows task-to-task logical connections to be established between a pair of connected PDP-11's/VAX's. This driver implements several interesting software mechanisms including non-standard AST's, driver maintained queues, and talk rundown support. It allows a network operation to be processed with less than 10ms software overhead.

* REAL-TIME BASIC-PLUS-2 TECHNIQUES UNDER RSX-11M

A system has been designed to acquire Quality Control inspection data from a shop floor using semi-automatic and automatic gauging methods. This system is written in BASIC-PLUS-2 and runs under the RSX-11M operating system. Data acquisition sections of this system operate at a higher priority and must be as efficient as possible. This talk will discuss techniques used to achieve this goal including RSX-11M Executive calls from BASIC-PLUS-2 and a performance monitor which allows one to determine which sections of BASIC-PLUS-2 code consume a significant amount of CPU time.

* INTRODUCTION TO RSX TASKBUILDER OVERLAY CAPABILITY

The Taskbuilder provides a means to reduce the memory and/or virtual address space requirements of a task by using tree-like overlay structures created with the Overlay Description Language (ODL). This session will describe the kinds of overlay structures: overlay trees, overlay data structures, and user-time routines. Also discussed will be the the Overlay Description Language, multiple tree structures, and various overlay loading methods.

* USING XDT FOR FUN

XDT, as documented, is a very practical tool and can be quite informative. However, there are some less obvious uses for XDT, such as: making hot patches to the EXEC, teaching system programming, device reconfiguration, and reorganizing memory. XDT is everything from an editor to a system monitoring routine. In this session we will try to demonstrate some of the less practical and more enjoyable uses of XDT.

* CONTROLLING THE ALLOCATION OF A SINGLE USER RESOURCE IN A MULTI-USER RSX SYSTEM

Following the conversion of our system controlling a Keithley D.C. parametric test system via the IEEE-488 bus from a single-user RT-11 to multi-user RSX-11M system, our user began to experience problems due to multiple test programs simultaneously accessing the test equipment. In addition, other tasks in the system were influencing the time delay between sets of measurments. A number of test programs which depended on fixed time interval between measurement values were failing. This paper will discuss the use the RSX-11M system directives to implement a scheduler which controlled access to the test hardware and task response time.

* AN RSX IMPLEMENTATION OF THE SEMICONDUCTOR EQUIPMENT COMMUNICATIONS STANDARD (SECS)

The SECS network protocol was developed within the semiconductor industry for the purpose of interconnecting semiconductor processing and testing equipment to implement computer-aided manufactoring (CAM) systems. This paper discusses the details of three implementations of subsets of the standard under RSX-11M. Paritioning of the protocol functions between drivers, applications

tasks, and ACP's will be discussed for each application.

* USING OPE FOR FUN AND PROFIT

  OPE, as documented, is a very pratical tool and can be quite informative. However, there are some less obvious uses for OPE, such as debugging device drivers, debugging tasks, fixing pool problems, and aborting unabortable tasks.

* RSX/IAS SPOOLER FOR THE CALCOMP 1055 PLOTTER WITH 907 INTELLIGENT CONTROLLER

  The standard calcomp plotting software has been modified to permit a user program to automatically queue the output files to a system disk. A friendly menu-driven dequeuing program was developed for an operator to submit the plot files to the Calcomp 907 Controller according to paper type/size and pen configurations. The entire package is written in F4P and Macro-11 and can be run under either IAS or RSX.

* AST'S AND SST'S IN AN OVERLAY ENVIRONMENT IN RSX-11M/M+

  The RSX-11M/M+ system provides two powerful tools in the Autoload Overlay and System Trap facilities. Unfortunately, these two features are very nearly incompatible. Our particular experience with this problem involves AST routines which call subroutines in shared library with memory-resident overlays. This paper examines the interaction between AST and SST traps and the overlay software. This examination includes overlay internals, techniques for eliminating the problems of overlays and AST/SST routines, and suggested applications.

* TABLE DRIVEN PARSING FOR LARGE GRAMMARS IN RSX

  THE RSX TPARS table-driven parsing routines are a useful set of routines which make the parsing of command lines and simple utility command grammars easy. The routines are very flexible and comprehensive, but are limited to the size of the associated state tables, about 100 words. When it is necessary to parse a larger grammer - such as an English language grammar for a command line interpreter - it becomes necessary to use a parsing technique which allows a larger grammar. This is most easily accomplished using the techniques of state-table parsing with an associated "dictionary" to allow look-up of grammar elements. This session will discuss and demonstrate the techniques used in state-table driven parsing. An implementation of a table-driven paser for a computer Dungeon game will be used as an example.

* SAMPLE FILE CONTROL - A LABORATORY DATA MANAGEMENT SYSTEM

  Sample File Control is a laboratory data management system implemented on a PDP-11/70 IAS V3.0 system in a file management language called INFORM. The design provides flexibility by placing important system variables in tables which each laboratory site may revise to meet its own needs.

* WHY SHOULD I KEEP THE SYSTEM LIBRARIES ON FLOPPY DISKS? - MEMORY RESIDENT DISKS FOR RSX-11

  With the increasing popularity and use of 22-bit machines, along with the decreasing cost of main memory, it is feasible to store libraries and work files in main memory as a "memory resident disk". This session discusses the implementation of a memory resident disk driver for system libraries under the RSX-11M operating system.

* VIRTUAL RECORD LOCKING IN RMS UNDER RSX-11M/M+

  The record level locking of RMS files under VMS is not available under RSX. Our warehouse inventory control system, with three dozen tasks accessing the same data base in real-time could not tolerate contention for a locked bucketful of multiple records. The solution was to implement a centralized data base access task that virtually locks records by RFA.

Atlanta RSX-11 Question and Answer Session

Ralph W. Stamerjohn
Multi-Tasker Editor

The RSX-11M/M+ Question and Answer Session was held on Monday evening at Atlanta. Bob Denney chaired the session. The following people from Digital were available at the session, and throughout the Symposium:

Steve Paavola	RSX, RMS Product Manager
Stevie Adams	Multiuser Tasks, Error Logging
Jill Angel	Documentation
Tim Day	SPR's, Autopatch, RMS, File System
John Franzini	SYSGEN, Batch, Accounting, Shadow Disk
John Gemignani	RMS, File System
Hai Huang	MCR, Executive, VMR
Leonid Kogan	BRU, Floppy Drivers
Jane Lawler	TKB, Executive, MAGM FCS, File System
Rich Perron	I/O Drivers, TTDRV, MTAACP
Craig Putnam	VAX/VMS AME, Error Logging, FCS, File System
Lee Siler	DCL, CLI Support, ICP

At the beginning of the session, Steve Paavola restated the current status of RSX-11M, loosely defined as mellow. A previous issue carried a note from Steve on RSX-11M's status. He also read the new SPR publication policy, which has also been carried in a past issue of the Multi-Tasker

Tim Day then mentioned four known problems with RSX-11M V4.0. First was the incompatibility problem between BRU V3.2 and BRU V4.0. See the SPR column for more on this.

Next, Tim mentioned the problem with some of the .BLD command files when using the RSX-11M V3.2 IND. He noted that FTB was shipped with some bad modules. The correct files will be on the Autopatch B kit. Finally, subfunction IQ.X does not work as documented, and will be retracted. Appearantly the subfunction bit being used in this case was already in use by another function and, until a new implementation is developed, there will still not be a way to get an immediate error return when trying to attach to an already attached device.

Then the question and answer session began. The following transcript comes from various sources: a transcript prepared by Gregg Merrell of Digital, the answer forms collected at the session, and the recording made.

Larry Solomon, Kulicke and Soffa

Q: I have 2 line printers in two locations, one of them is remote. Is there anyway under RSX-11M+ I can implement automatic routing. That is, the user would give a local assignment that would make /SP and PRINT go to correct printer.

A: No answer at session. Some users indicated they have done something along these lines.

Joseph Sventek, Lawrence Berkeley Laboratory

Q: With RSX-11M V4.0, we finally have FCSRES support on SYSGEN. Unfortunately, the support is only for PLAS FCSRES, conditional on ANSI magtape support. Could you please work out modifications to SYSGEN to permit the tasks to be built with the FCSRES of my choice and make available on a future Autopatch kit.

A: Will look at it.

James Johnson, Houston Chronicle

Q: Are there any plans for a fault tolerant RSX-11M type system?

A: No, not in the RSX area.

Dale DeMott, Caterpillar Tractor Company

Q: With the advent of DCL and flying installs, is there anything planned to remove the 80 character limit on MCR commands or support for continuation lines at MCR level.

A: No.

Anthony Scandora, Science Application

Q: Under RSX-11M V4.0, when an F4PRES of 4K links to a 4K FCSRES, it doesn't work as the second region isn't mapped.

A: This will fix be fixed.

C: A workaround is to build F4PRES to FCSRES. Then ZAP F4PRES.STB to remove the reference to FCSRES. When building the target task, use two LIBR= statements.

Anthony Scandora, Science Application

Q: The distributed F77 F4PRES ODL won't build.

A: Move SAVR1 to the root as the OTS overlay routines need it.

Jerry Williams, Computer Science and Applications

Q: During SYSGEN, five UMR's (7 if DH-11 is present) are statically allocated. What are these used for in the system? Can I use them for myself?

A: They are used to overmap the EXEC so vectors and pool are available for devices that need them (such as TS11's).

Alan Frisbie, Flying Disk System

Q: Why can't VAX/VMS read RSX-11M V4.0 DSC tapes?

A: Probably because it is RSX-11M DSC from V3.2.

Denny Walthers, American McGaw

Q: Why does BYE reset the terminal to /HDX (no FDX)?

A: Development is considering a taskbuild option to BYE that would allow the user to reset characteristics to whatever desired. But this is true for now.

Ken Robinson, AMAX Copper

Q: Under RSX-11M+, I am having problems with batch. Jobs die without errors listed on the log file, but the console indicates BPR errors. How come?

A: Unknown, but RSX-11M+ V2.0 has queue manager fixes.

Franklin Reynolds, Intermetrics

Q: What kind of problems will I have moving privileged tasks from M3.2 to M4.0?

A: Any reference to UCB's and TCB's should be checked. The release notes have a very complete description of all pertinent executive modules and data structures.

Robert Thomas, A.S. Thomas

Q: I am having trouble rebuilding M3.2 Fortran tasks and commons under M4.0. In particular, I get conflicts from the .PSECTs $$$RTS and $$$AVL, the .PSECT .PDATA is zero length, and I need to rework my previously good ODL's.

A: Ignore the .PSECT attribute error message or use the /NM (no message) switch to suppress such diagnostic messages. (The .PSECTs were changed from RW to RO for use under M+ I/D space and multiuser tasks.) No idea on .PDATA. The ODL problem belongs to Fortran.

Guy Lauten, Friden

Q: On an Emulex tape cpntroller at 75IPS I get write CRC errors from BRU. The media is OK and FLX and PIP do not seem to have errors. What does BRU do that FLX and PIP don't.

A: BRU is very fast and use large buffers, especially when compared to FLX and BRU. Move the controller furthest from the processor to try to help the data late errors.

Jay Nelson, HERCO

Q: When reading the driver for RL's, I found a test that spins on a bit in the seek request done section. Is that OK?

A: Yes, as it is only done on control functions, not move functions.

Richard Bielak, ITC

Q: I tried to move a non-physical device driver from M3.2 to M+1.0 but could not CON it ONLINE.

A: CON needs to get a valid response on the bus from the specified locations. Use a vector and CSR of 0 so that memory will respond for you to keep CON happy. Look at NLDRV as an example.

Rick Royston, Dow Chemical

Q: Does leaning on the arrow keys of a VT100 still kill M4.0, M+2.0, VMS3.0?

A: It is not good for any of them. M+ can survive it a little better due to use of secondary pool for MCR command line buffers.

Mark Weston, City of Gainesville

Q: Under what circumstances does indirect .SET QUIET not function. I am aware of requirement of parent task offspring support.

A: In the release notes for M3.2, it says to patch 2 files and then rebuild IND.

Minaz Ladha, General Motors Institute

Q: Must I apply each autopatch in turn?

A: No, each one is complete in and of itself.

Burt Harris, JHU/APL

Q: Why have my RSX-11M V3.2 SPR's regarding user mode diagnostics been answered saying that UMD are not supported in RSX-11M V4.0 when the Software Product Description for RSX-11M V3.2 calls for them? Why are they not supported in RSX-11M V4.0?

A: UMD is not supported under RSX-11M V4.0 because no one could be found to support them under RSX-11M V3.2. Digital thinks that IOX should provide UMD functionality. Users replied that an exerciser cannot perform test as thoughly as a diagnostic. Digital stated they would rather increase the functionality of IOX rather than support UMD.

Ted Jackson, Union Carbide

Q: Under M4.0 when writing an ANSI tape on a TU10, I get an extra HDR1.

A: Have not heard of such a problem. Please submit SPR.

Joe Whatley, A.C. Nielson

Q: Is there a problem with the task activation priority threshold with PMT (pool monitor task) under M4.0? When set to 51, a task of 50 will run when a pool low condition is reached.

A: Not aware of this problem.

Jim Riccio, RCA

Q: Do I need to reinstall/rebuild RMS from M3.2 to M4.0?

A: You can copy over tasks, and reinstall the object libraries, particularly those in SYSLIB.OLB.

R.E. Grandle, NASA

Q: I get an error message from DCL of 'Task Not Installed' even though I build it for flying installs.

A: Rebuild DCL with an optional patch in the build file. Note that the M4.0 GBLPAT is in the wrong place, it should be moved to the end of the .CMD file. It will be fixed on Autopatch 'B' (due to ship 4 months after M4.0 first customer ship).

Thomas Gojan, Western Union Telegraph

Q: Online DSC disk to tape with verify hangs all drives until rewind is done on DSC tape.

A: M4.0 implements overlapped rewinds which fixes the problem.

Charles Goodpasture, Trasnco

Q: Are there any problems when running DSC from disk to tape or tape to disk?

A: No problems known.

John Vilandre, University of Minnesota

Q: Will we be able to link an F4PRES to FCSRSL without "tricking" the task

builder in RSX-11M+ V2.0.

A: Yes.

Louis Stoll, Zia Company

Q: What are the plans for an M4.0 compatible AME on VMS?

A: The schedule is unclear as big changes are needed. Note, that the RSX group now owns the VMS AME and intends to get as close to RSX-11M V4.0 as possible.

Barton Bruce, Cambridge Computer Associates

Q: Does multi buffering/big buffering work OK under M4.0? Specifically, does Fortran have a problem when closing a (TT:) unit on which a control-Z was received?

A: No problems reported at this time (ED: see SPR column for Fortran problem.)

Allen Bennett, Clark Handling Systems

Q: I use RMS under RSX to read with an alternate key by RFA, but it seems to be going back to the primary key when I do a FREE and then later return to the file.

A: Use a random FIND on the alternate key to restore the alternate key context. RFA by definition establishes primary key.

Larry Solomon, Kulicke and Soffa

Q: I am trying to send multiple sends to a single task, but seem to loose some. They are being dequeued by use of AST's.

A: When data is received, an AST is posted only if the receive queue was empty at the time the data arrived. Therefore, after doing the SRDA$ and before doing the ENAR$, do RCVD$'s to clean out the queue. In the AST routine, keep looping through a RCVD$ until getting an error (IE.ITS) before exiting the AST routine (ASTX$S). Also, do RCVD$ on task startup and after declaring AST address to remove anything already sent.

Bryant Moriarty, The Record

Q: I fixed BRU in an system along with F11ACP and MCR for standalone backup, but it doesn't work right the second pass.

A: Serial reusability is not guaranteed for distributed tasks.

C: Please make that point very clear if that is really true.

Ken Robinson, AMAX Copper

Q: Can I use user mode I/D space under RSX-11M+ V2.0 with COBOL and RMS?

A: Not sure about COBOL. RMS V2.0 may have. F77 V4.0 works.

Jerry Koontz, Burroughs Wellcome

Q: A number of the devices I hang off of our system use the TT: driver as they are RS-232 devices. I would like to eliminate some of the terminal specific overhead of the driver while retaining the executive specific functions. Could the TTDRV code be split so that I can eliminate the terminal specific portions?

A: There is no room unless we retain the current structure.

Bobby Frizzell

Q: My user written driver under RSX-11 V M3.2 uses error logging and I want to move to RSX-11M+.

A: Change to the error logging interface for M4.0/M+2.0, but should be much better for adding user drivers.

William Dorfmann, Eastman Kodak
Q: How can I get my RSX-11M V3.2 indirect command files to recognize exit status?
A: Rebuild IND to use parent-offspring tasking.

Kitty Bethe, Banker's Trust
Q: I issue many IO.RTT's (read with special terminator). It sure would be great if I could specify the termination table on an IO.ATT instead of each IO.RTT. (It would also let me use IO.RPR which I currently can't do.)
A: Please submit a suggestion SPR.

Gary Maxwell, U.S. Geological Survey
Q: Is there any way to restrict access to either Batch or Print queues in RSX-11M-Plus?
A: No, but we will look into it.

Larry Baker, U.S. Geological Survey
Q: EDT tries to open its workfile with the NOSPANBLOCK attribute. If the input file has records of 512 bytes, EDT goes into an infinite loop because it has a two byte header and 514 byte records always span 512 byte blocks. EDT keeps allocating blocks to the file until the disk fills.
A: This is an FCS problem and it has been SPR'd.

Anthony Scandora, Science Applications
Q: When will RMS convert VMS decimal version numbers to octal? RMS BACKUP is the only supported way to transfer files between RSX and VMS, so it is important to get it right.
A: Fixed in RMS V2.0.

Denny Wacthers, American McGraw
Q: When will cluster libraries work with RMS?
A: RMS V2.0. (Estimated 6-12 months away.)

Roger Jenkins, Wycliffe Bible Translators
Q: An RMD-like print queue display would be nice, especially for our operators.
A: Will consider.

Alan Frisbie, Flying Disk Systems
Q: My foreign RX01-compatible drive causes ACF to hang under M4.0.
A: ACF and device drivers are very picky that the device look exactly like the DEC device for error logging purposes.

Richard Ceci, Advance Technology and Test
Q: Can I run any line of a DLV-11J above 2400 baud? I seem to be having a problem. The TSC suggests 2400 max.
A: You could loose interrupts if it is at the end of the bus.
C: You should reconsider the standard configuration of the MINC-23 systems.

Allen Bennett, Clark Handling Systems
Q: Can the modules of RMS V1.8 be split in any way to support clustering, I/D space, or supervisor mode? I really need the space.
A: No. Clustering is not possible because RMS passes data on stack. But RMS V2.0 will be able to be clusttered.

Terry Medlin, GEJAC

Q: Have all the bugs in COT been fixed in RSX-11M V4.0?
A: COT has been completely rewritten in RSX-11M V4.0. It now buffers messages internally and should not loose messages.

Robert Thomas, A.S. Thomas
Q: I do not have a hardware contract on my system. I would like to see publication of ECO's and FCO's that are required to let the software run properly.
A: A hardware subscription service is available for those who wish to do self-maintenance.

Franklin Reynolds, Intermetrics
Q: How portable is RSX-11M software to the PC350?
A: Tasks are reasonable portable through the use of the tool kit. The P/OS looks like modified RSX-11M+ with RMS standard, no FCS or MCR, ODS1 currently and soon ODS2.

Jay Nelson, HERCO
Q: I want to do a single generation of RSX such that I can use the pack on both an 11/34 and an 11/44. What precautions should I take?
A: If the system is gened for the 11/34, it should work fine on the 11/44. Note that you will only be able to use up to 124KW on the 11/44.

Marylin Forrest, Burlington Industries
Q: My task needs a command line and a reschedule interval, but SPAWN$ doesn't allow a reschedule interval.
A: The CO: device which is where all clock queue tasks run isn't logged in. Only a logged in terminal allows spawning due to passing of command lines. (Much discussion of alternatives.)

Jess Goodman, ACCU-Weather
Q: IO.ATT to a device which is already attached hangs.
A: IQ.X was supposed to fix the problem, but does not. A new implementation will be developed.

Burt Harris, JHU/APL
Q: Are there any plans to allow inline data in indirect command files as VMS command files permit by reading from SYS$INPUT?
A: No plans.

Bobby Frizzell, Dalas Power and Light
Q: Why does RSX-11M+ not support the DRS-11 hardware?
A: RSX-11M+ only implements those newer I/O devices felt to be reasonably common. In particular, it does not support DRS, DSS, ICS, ICR.

James Johnson, Houston Chronicle
Q: We are sticking with M3.2, and bought autopatch so we would get autopatch 'E'. Since M4.0 autopatch 'A' is the current version, how can we get M3.2 'E'?
A: See if you can get a copy from your local office, or if they can order a copy of the now out-of-date autopatch 'E'.

Barton Brice, Cambridge Computer Associates
Q: Why not ship BRUSYS.SYS along with BRU64K and DSC64K on M4.0 as most 11M users have 96K systems today.

A: Will consider.

Steve Gabelnick, Argonne National Lab
Q: My foreign RM-type disks which are larger than the DEC drives cause the system to not boot.
A: The DR: driver looks at the device size to select the device geometry at boot time. The size is fixed at gen time. RM's require the exact size for each drive in the system, as do RK06/7's.

Patrick McHugh, NCR
Q: FLX chokes if the RX02 diskette is the wrong density.
A: It has been patched to retry on the other density. See Autopatch E.

Gary Maxwell, U.S. Geological Survey
Q: We don't like the /SP switch on MAC, TKB, etc. Many of the utilities come with optional patches in the build command files to disable spooling, but MAC and FTB require an SPR to find the patch location for the switches.
A: Either remove PRT... or submit an SPR.

Kitte Bethe, Bankers trust
Q: Under RSX-11M V3.2, many utilities did not exit with status, in particular INI, BAD, and FMT. Do they under V4.0?
A: BAD and FMT do. We will check INI on the demo systems.

Wade Scannell, Athena Systems
Q: We like many of the features in the TDX tasks floating around. Why are some of them not in the M4.0 TDX?
A: The specific TDX distributed with M4.0 is not supported, but is a template that you may customize for your own needs. It has code to show how to do most types of commands.

Larry Baker, U.S. Geological Survey
Q: ABRT$ processing is done at normal task priority rather than elevated priority. This can result in rundown taking a very long time for some tasks such as those stopped for I/O. Can this be changed?
A: (Much discussion) We will try to publish the nature of task aborting.

Louis Stool, Zia Company
Q: The Professional 350 will implement ODS-2. Any plans for RSX?
A: No.

Jay Nelson, HERCO
Q: If multiple tasks are waiting for the same global flag, only the first seems to get resumed.
A: It should work for all tasks waiting.

Tom Gaps, Tektonix
Q: My M+ system is gened for a DH-11 with DM-11 modem support when there is no DM-11, the system crashes if I set the line /NOREMOTE, even if the device is CON'ed OFFLINE.
A: Set the CSR to a memory address that is there to fake out the TTDRV.

Anthony Scandora, Science Applications
Q: How can I change the defaults on DCL commands?
A: The sources are distributed with all tables coded in MML (Macro Meta

Language). Refer to Chapter 12 of the manual.

Gary Maxwell, U.S. GeologicalSurvey
Q: Why is MCR overlaid only?
A: It can be flattened in RSX-11M+ V2.0.

Charles Ham, E.I. Dupont
Q: Why is the error logger report program called RPT? It is such a natural name for a report task that probably hundreds of sites have already used it.
A: It seemed as reasonable for us as it did for you to use it.
Q: When I try to extend a big directory on a full disk, the disk gets scribbled.
A: Please SPR.
C: A number of people remarked of the same problem, and also the inability to recreate it. If anyone has had this problem and can document how they did it, contact the Multi-Tasker.

Toronto RSX-11 Question and Answer Session

Lorne LaFramboise

STELCO Inc.
Hamilton, Ontario

A Question and Answer session was held at the Toronto DECUS Symposium on Thursday, February 11, 1982. The session chairman was Mike-Blake Knox from Northern Telecom. Digital panelist were Steve Paavola and Tim Martin.

RAY LINSEMAN (AEL MICROTEL, 100 Strowser Blvd., Brockville, Ont.)

Q1. How can we submit feature requests for releases of RSX-11M? A1. Use the SPR and check off Suggestions.

Q2. What is the 11/74, why isn't it being shipped, could this change in the future? More generally can we expect to see it in the upper end of the PDP-11 family. A2. Project cancelled. For product reasons 2 years ago, with 80 processors built but thought cost to high. There are two new processors coming,but below the VAX. PDP-11's are not dead.

Q3. Using M V3.2 on the VT100 hitting the RIGHT CURLY BRACKET (IE. SHIFT RIGHT SQUARE BRACKET) on a command line to MCR (eg. PIP) causes processing of the line to that point instead of echoing the character. Is this a hardware problem on the VT100 or software problem. A3. Problem in definition of code /176 . Do a SET /LOWER and use the CAPS LOCKS.

Q4. What attempts are being made to make online activities easier, eg. for on-line Syssgen transfer account file to target disk based on yes reply to prompt or on-line diagnostics for DL11-E and DUP-11? A4. Not much yet but will be addressing it in new releases and looking for input.

Q5. Is it true that the 11/70 did not obtain FCC approval? If yes, what is the future of the 11/70? A5. Have verified, does meet the FCC rules. Various strategies are being investigated to handle expected 200 more systems. Cutoff date is OCT 1983 for manufacture, but expect deliveries after this.

DEC. FCC requirements specify that 'All electronic data processing equipment manufactured after OCT 1983 must pass testing for limited amounts of radiation; electronic and magnetic'. DEC is testing old products to be manufactured after OCT 1983 and doing modifications to see that they conform. DEC could make 11/70 conform and will conform if shipped after OCT 1983.

Q6. Is anything in the works with higher density memory boards to increase the memory of PDP 11/44 from 1 meg to 4 meg? If yes would changes be required to M+. A6. 11/24 use 16K bit chips. 11/23+ use 64K bit chips. It is reasonable to expect it will come out for 11/44. (DEC standard response #4) Do not expect any changes would be required for M+.

BILL KORENDYK (ALBERTA RESEARCH COUNCIL, Edmonton, Alberta)

Q. When is the next release of DECNET-11? Will it include a supported multi user RVT? A. (How do I answer, I know more than I can say) DECNET is now in field test, but no schedule for release is determined. RVT will be supported but will not be multi-user.

RICHARD JONES (NORTERN TELECOM, 44 Peel Center Dr., Bramalea, Ont)

Q. 11M V3.1 didn't force close and reopen of the (TI:) when EOF (^Z) was typed. 11M V3.2 forces us to close and reopen files after (^Z) is typed. How does 11M V4.0 handle EOF? A. In Fortran under V4.0 one must rewind the terminal, to reset the EOF condition created through FCS when the terminal driver returns an EOF. You don't have to close the file.

L. A. CRAGG (TEKLOGIX INC., 1199 Fewster Dr., Mississauga, Ont)

Q. How do you set a cursor control input from a VT100 in ANSI mode. A control key generates ESC, x, y but RSX does not allow the 2nd character (y)? A. Set the terminal to enable Escape Sequences. This allows characters following the escape to be allowed into the terminal buffer.

ARTHUR J. GEORGE (SIMPSON-SEARS, 222 Jarvis St., Toronto)

Q. How do you enable and use block locking for FCS files? A. Documentation of use can be found in Los Angelos (1981) Proceedings.

LORNE LAFRAMBOISE (STELCO INC., Hilton Works, Hamilton, Ont.)

Q1. How do I spool to more than one LP device , that is if I have two LA120's? A1. Install $LPP as INS $LPP/TASK=TTPn Then to spool to that device specify PRI TTn:=name

Q2. Will SHADOWING RECORDING be available for RSX-11M V4.0 like it is on RSX-11M+ V1.0 ? A2. No. Shadowing recording is with M+ only. The amount of overhead and the data structures are significantly complex. Secondary pool where large data structures are kept is required for shadowing recording.

PHIL OLYNYK (BELL CANADA, 2 Bloor St. W.,Toronto)

Q1. How much life does RSX-11M have, in terms of years of support? A1. New releases will be planned. DEC will stabilize 11M, fix bugs, add new device drivers; for new functionality see M+ new versions.

Q2. Does anyone know anything about filling up TTDRV pool and system pool with CR/LF when the buffer is set small, and output is pending? Pool fills up with CR/LF. A2. Known bug. M V4.0 puts minimum size to SET /BUF command.

BRYAN SMITH (DUPONT CANADA INC., BOX 5000, Kingston, Ont.)

Q. The documentation of RSX-11M V3.2 Serial Despooler says that, when passing a file to a serial despooler which has been sened for no deletion, you can nevertheless obtain deletion if a certain argument is negated. (Can't be sure of which without seeing the manual) Can't get this to work. What does negated mean? ie. -n; n.or.Bit 7; n.or.Bit 15 or what? A. No answer at this time.

BRAD TINNEY (CANADIAN HYDROGRAPHIC SERVICE, Burlington, Ont.)

Q1. Does V4.0 have a time stamp on COT that includes the date? A2. No, when time stamp rolls around it puts the new date into the log file.

Q2. Explain the mechanism where most installed tasks hold off the MCR prompts until done, but others such as ACNT doesn't hold it off. A2. When saying RUN [installed task] then the RUN function dispatches the task and MCR command RUN is over and the prompt appears. This a feature not a 'bug'.

JOHN BRODIE (MILLER COMM. SYS. LTD., 300 Legget Dr., Kanata, Ont.)

Q1. Does the full duplex- terminal driver put solicited input only into the typehead buffer if output is simultaneously in progress on ths same line? A1. No answer at this time.

Q2. MCR prompts sometimes are lost if CTRL/S is entered just as (e.g.) PIP TI:= file.* completes. A2. Could be problem with VT100.

Q3. Why does KED sometimes repaint the screen when scroll would do? (e.g. duplicate lines) A3. Submit "FYI" SPR.

DOUG BUCHANAN (DOFASCO, Hamilton, Ont.)

Q1. V3.2 full duplex terminal driver does not support asynchronous I/O. How can I get around that problem. A1. If you want the QIO to be asynchronous make the task non-checkpointable issue the QIO, then don't make the task checkpointable until the data transfer is complete.

Q2. With Fortran IV-Plus (not sure of the version) if you don't use the /CO switch, which allows excess continuation lines, for the compiler in some circumstances we hang the system and must ABO F4P. A2. May be unique to the old version of F4P.

GERRY D. MINTZ (UNIVERSITY OF TORONTO, 4 Taddle Creek Rd., Toronto)

Q1. What is the status of DEC Standard Runoff, with regard to 11M and 11M+? Will it be distributed? I understand it is written in BLISS and is very large. A1. It exists but we can't have it.

Q2. Describe the hooks into the terminal driver included to support FMS-11 field validation, are they accessible via QIO's? A2. This is implemneted with the subfunction bits but is not yet documented because DEC hasn't decided what support will be neccessary, nor is the FMS Development Group certain of the final architecture they will use.

MIKE BLAKE-KNOX (NORTHERN TELECOM, BOX 3000, Brampton, Ont.)

Q. Will the RSX-11M V4.0 Autopatch kits follow the correction UIC philosophy documented in the release notes and will one be able to apply them to any existing disk? A. Surprised this is being done. Will look into what can be done to make it easier to merge corrections and account for local executive modifications. Internally [x,45] are COR files and [x,65] are PAT files.

SHERMAN O'DONNEL (ROYAL MILITARY COLLEGE, Kingston, Ont.)

Q1. Is there a method to use my dial up-modem on a DZ11 as a terminal to another computer? Can I then set a VT100 on a different teminal line to act as the terminal calling out on the modem? A1. Yes, if you have a program that reads from TT1:(terminal) and writes to TT2:(modem) and back. Also SET TT2:=NOREMOTE and SET NOECHO on the modem.

Q2. Can I use just a reassignment to do the above? A2. No. Use DECUS program Duplex #11-456.

Q3. What does the utility FMT do when executed at an RX02 drive? Our system just hangs and must be rebooted if the disk is labelled, 'unformated soft sectored, double density'. A3. No answer at this time. Submit SPR. Send floppy. DEC not sure what FMT does for bad sectors on the floppy.

Q4. Did you say that there was a new facility in M V4.0 that will allow the user to specify the length of forms? Also is there a 'DEC approved' program, or switch to set the EVFU on the Printonics line printer, or is this a Queue Manager problem? A4. Yes. V4.0 will allow users to specify form length. Different endings to different files cause varying actions at the line printer. There is no standard EVFU program from DEC. Check the PLXY driver option at task build time.

MARK JOHNSON (ALBERTA RESEARCH COUNCIL, 1135-87 Ave., Edmonton, Alberta)

Q. Since the M developers are now responsible for the VMS AME, will it then be supported for M Syssen on a non-dedicated disk? I have a RM80 disk. A. RSX-11S Syssen is the only one supported for VMS. RSX-11M has not been tested. Distribution is on DSC tape, so it will initialize disk, but it can then be used. There are no plans for ODS-2 support during M Syssen.

BRIAN A. KERR (UNIVERSITY OF TORONTO, 215 Huron St., Toronto)

Q1. Ocassionally the interrupt enable bit on console keyboard set cleared. A1. Write a small task to periodically reset the bit.

Q2. Some RSX Utilities overmap the I/O page ,why? IE., $ACNT and the Indirect command processor. A2. It's a priviledesed task. Just ignore the message, tasks map the I/O page space but don't use it.

JOHN HOGE (DUPONT CANADA, Maitland Works, Kingston, Ont.)

Q1. On M V3.2 it was convenient to Spawn HELLO and Logon CO: so that it could spawn indirect on CO:. On M+ Spawning HELLO on CO: crashes the system? A1. Yup! Fixed on M+ V2.0 .

Q2. When do I get into trouble using a DZ11 rather than a DH11? What performance do I loose? A2. DH11's are suggested for remote lines so it doesn't have to go through modem code.

Q3. Can EDT and KED benefit from using a DH11? A3. Repainting the screens with EDT and KED would benefit from a DH11.

Q4. Our M+ system 'goes to lunch' (no prompts) at random intervals and comes back after 5 to 30 seconds. RMD shows nothing much is running. A4. Have experienced the problem before, check Software Dispatches. Believe tasks wipe the I/O status block, waiting successful completion, however the system never sets the I/O status block back to the task.

Q5. On an 11/70 with RM03 and TS11, BRU still gives us problems when doing online backup of the whole disk. BRU requests mounting of multiple tapes forever. Also opening the door on the TS11 crashes the system. A5. Digital will do more testing in house with TS11 tapes for end of tape position. Apply M and M+ patches from the Software Dispatches to BRU.

PETER TIMPF (STELCO INC., Lake Erie Works, Nanticoke, Ont.)

Q. Does DEC have plans to convert existing utilities to what appears to be a new convention ie. specifying INPUT FILESPEC= OUTPUT FILESPEC as in BRU,DSC? A. This a difference between DCL and MCR. No there are plans to do this.

JOHN BOUFFORD (MINISTRY OF RESOURCES, Maple, Ont.)

Q1. It appears as if the high order byte in text files is not used. Is this so? There is a percent sign in before every character when the file is displayed using DMP. A1. Each byte is being used. The Percent sign is used by DMP to imply that the character is lower case.

Q2. Is it possible to remove the CO: driver in a V3.2 system. A2. Yes. CO: logging will be of course be disabled.

BRU Sorting Bug - A Followup

Carl T. Mickelson

Goodyear Aerospace Corp
Akron, Ohio 44315

In an SPR submitted to DEC in May, and published in the Multi-Tasker in July 82, I outlined a bug-fix to the LBNORD module of BRU. Since submitting the SPR, I have had two telephone discussions with the DEC software maintainer responsible for BRU and have received his SPR response. While admitting that my fix was correct, DEC claimed, rightfully, that it was incomplete. It seems that the quicksort algorithm as implemented in the LBNORD module always stacked the left partition after dividing the sort list. A properly implemented quicksort stacks the larger (left, or right) partition. Due to this error, it is still possible under rare circumstances to cause a data dependent stack overflow that will terminate BRU execution. Further, it can not be guaranteed that the retrieval pointer list returned by the incorrect routine is sorted under all circumstances. This means that any correction made to implement a proper quicksort could lead to incompatibilities, because of out-of-order data on tape, between a revised BRU and existing tapes made prior to the LBNORD correction.

DEC's solution for this problem is to leave LBNORD as it is, to add a new, revised sort routine to BRU, and to mark the save tape format as to which sort routine was used to make the tape. In this way, the revised BRU program can determine which sort routine to use to restore any BRU tape, regardless of the version that created it. A forthcoming (no date supplied!) Dispatch Article was promised to outline all the corrections necessary for this approach. They are

said to be extensive. In the meantime, I have developed a patch to LBNORD for V3.2 BRU, and a companion but untested patch for V4.0 BRU, that stacks the larger quick-sort list partition to correct this problem. We are using this revised BRU at our site to avoid the possibility of stack overflow terminations. Since we use BRU primarily for full-reel full-saves of our disc on a weekly basis, after one month recycling our save tapes, the BRU tapes will no longer be potentially incompatible.

It should be noted that while this patch also effects V4.0 BRU the BRUBLD.CMD file built during a V4.0 SYSGEN applies a GBLPAT to LBNORD that restores the original out-of-order count contained in LBNORD as distributed for V3.2! This effectively cancels Article 5.1.17.9. The GBLPAT should be modified at any site that adopted the technique of this article, since the change causes LBNORD to skip sorting the data, hence avoiding the sort errors, and changes the order of data on a tape so that it may not be restoreable by a different version of BRU. Note also, that the GBLPAT should not be removed from the BLD file, but only changed if necessary, since the V4.0 distribution of LBNORD uses an out-of-order count of 40. and not 20. as distributed in V3.2. If the cumulative patch for V4.0 presented here is applied the GBLPAT can be removed.

Copies of the LBNORD patch files for V3.2 and V4.0 are included here with both errors fixed. Applying the new patches for V3.2 BRU is the same as the procedure published in July, except for the following line:

```
PAT LBNORD.OBJ;2=LBNORD.OBJ;1/CS:10755,LBNORD.POB;1/CS:33036
```

The patch for V4.0 given here is slightly different in patch locations than the V3.2 patch due to other changes DEC made in the baseline version for V4.0. Since we are still running V3.2, I have not been able to try this patch on a V4.0 BRU.

```
        .TITLE  LBNORD          ; FOR V3.2 BRU
;
; MODIFICATION:
;
;       01.2 -- FIX SORT LOGIC ERROR IN LBNORD (21 MAY 1982)
;               THIS IS PART OF THE REAL FIX FOR 5.1.17.9
;               C. T. MICKELSON, GOODYEAR AEROSPACE CORP.
;               AKRON, OHIO 44315 (216) 796 - 2388
;
;       01.3 -- FINISH CORRECTION TO QUICKSORT BY STACKING LARGER
;               PARTITION OF LIST RATHER THAN ALWAYS THE LEFT PARTITION.
;               THIS ELIMINATES POSSIBILITY OF A STACK OVERFLOW ERROR,
;               BUT MAY MAKE PAST BRU TAPES INCOMPATIBLE.
;               C. T. MICKELSON, GOODYEAR AEROSPACE CORP.
;               AKRON, OHIO 44315 (216) 796 - 2388
;
        .PSECT
 .BLK.=.

        .IDENT  /01.2/
 .=.BLK.+436
        DEC     $QSTAK+200.(R5)

        .IDENT  /01.3/
```

```
 .=.BLK.+12
I:      .BLKW   1
J:      .BLKW   1
ISTACK: .BLKW   1
JSTACK: .BLKW   1
 .=.BLK.+200
LOOP:
 .=.BLK.+362
        JSR     PC,PAT013
        NOP

 .=.BLK.

        .PSECT  $$PAT
PAT013:
        MOV     I,R0
        ADD     R0,R0
        SUB     ISTACK,R0
        CMP     JSTACK,R0
        BGE     10$
        MOV     $QSTAK-2(R5),$QSTAK(R5)
        RTS     PC
10$:
        TST     (SP)+           ;CLEAN UP STACK
        MOV     $QSTAK+198.(R5),$QSTAK+200.(R5)
        MOV     I,R0
        DEC     R0
        MOV     R0,$QSTAK+198.(R5)
        ADD     #2,R0
        MOV     R0,$QSTAK(R5)
        JMP     LOOP

        .END

        .TITLE  LBNORD          ; FOR V4.0 BRU
;
; MODIFICATION:
;
;       01.3 -- FIX SORT LOGIC ERROR IN LBNORD (21 MAY 1982)
;               THIS IS PART OF THE REAL FIX FOR 5.1.17.9
;               C. T. MICKELSON, GOODYEAR AEROSPACE CORP.
;               AKRON, OHIO 44315 (216) 796 - 2388
;
;               ALSO, REMOVE "FIX" OF 5.1.17.9
;
;       01.4 -- FINISH CORRECTION TO QUICKSORT BY STACKING LARGER
;               PARTITION OF LIST RATHER THAN ALWAYS THE LEFT PARTITION.
;               THIS ELIMINATES POSSIBILITY OF A STACK OVERFLOW ERROR,
;               BUT MAY MAKE PAST BRU TAPES INCOMPATIBLE.
;               C. T. MICKELSON, GOODYEAR AEROSPACE CORP.
;               AKRON, OHIO 44315 (216) 796 - 2388
;
        .PSECT
 .BLK.=.
```



```
        .IDENT  /01.3/
 .=.BLK.+56
        MOV     #20.,R2         ;Remove "FIX" of 5.1.17.9

 .=.BLK.+472
        DEC     $QSTAK+200.(R5)

        .IDENT  /01.4/
 .=.BLK.+12
I:      .BLKW   1
J:      .BLKW   1
ISTACK: .BLKW   1
JSTACK: .BLKW   1

 .=.BLK.+234
LOOP:
 .=.BLK.+416
        JSR     PC,PAT013
        NOP

 .=.BLK.

        .PSECT  $$PAT
PAT013:
        MOV     I,R0
        ADD     R0,R0
        SUB     ISTACK,R0
        CMP     JSTACK,R0
        BGE     10$
        MOV     $QSTAK-2(R5),$QSTAK(R5)
        RTS     PC
10$:
        TST     (SP)+           ;CLEAN UP STACK
        MOV     $QSTAK+198.(R5),$QSTAK+200.(R5)
        MOV     I,R0
        DEC     R0
        MOV     R0,$QSTAK+198.(R5)
        ADD     #2,R0
        MOV     R0,$QSTAK(R5)
        JMP     LOOP

        .END
```

I would now like to turn to an analysis of the LBNORD patch, .IDENT /1.01P/, distributed at the DECUS symposium in Atlanta, just after the release of V4.0. The patch is duplicated here for convenience:

```
        .TITLE  LBNORD
;
; COPYRIGHT (C) 1982
;
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS. 01754
;
; MODIFICATIONS
```

```
;
;       1.01P - PATCH TO FIX INCOMPATIBILITY PROBLEM BETWEEN
;               BRU VERSIONS 3.2/1.0 AND 4.0/2.0
;
        .IDENT  /1.01P/

        .PSECT

 .BLK.=.
CNT=.BLK.+60

 .=.BLK.+46
        CALL    PAT1

 .=.BLK.

        .PSECT  $$PAT
$$PAT=.
PAT1:   MOV     NUMRP,R0
        BIT     #KY.T2D,$QUAL3  ; CHECK FOR RESTORE
        BEQ     10$             ; NO, EXIT
        BIT     #CF.NST!CF.SRT,$CFLAG ;IS IT A 4.0 TAPE?
        BNE     10$             ; YES, EXIT
        MOV     OLDPAR,CNT      ; FIX OUT OF "OUT OF ORDER" RP'S
10$:    RETURN

OLDPAR::.WORD   24

        .END
```

This patch is reputed to solve the incompatibility problem between mag tapes produced between the V3.2 and V4.0 BRU's. The address CNT at .BLK.+60 points at the immediate value in the second word of the two word instruction MOV #40.,R2. This instruction is used to set the initial out-of-order threshold for performing a sort of retrieval pointers while processing a mag tape. As distributed in the BRU V4.0 library, this out-of-order count is set at 40(10). The subroutine call patched into the routine at .BLK.+46 (CALL PAT1) is used to determine whether this out-of-order count is to be modified to the value in OLDPAR (defaulted to 20.) when restoring a V3.2 BRU tape. The actual value used for the content of OLDPAR is determined by

```
GBLPAT=SCNDIR:OLDPAR:nn
```

which must be added to BRUBLD.CMD before taskbuilding V4.0 BRU.

The problem with this patch is that it prevents BRU from remaining serially reusable. Consider the following scenario:

1. Use BRU to restore a V3.2 tape to disc. The patch, when executed, will change the MOV #40.,R2 instruction to a MOV #nn,R2 instruction so that the V3.2 tape can be read.

2. Now, if the same invocation of BRU is used to create a new mag tape from disc, the tape format will be marked as a V4.0 tape, but the tape will be written using an out-of-order count for a V3.2 tape! This occurs because the patch provides no method for restoring the MOV #40.,R2 instruction. Note that if BRU used disc resident overlays to reload the correct instruction it could not be used to restore unmounted system discs, and memory resident overlays are not reloaded and so, once changed, remain modified.

3. The unsuspecting user now terminates BRU and archives the new mag tape. At some future time, this tape is remounted and a restore operation is to be performed. Now the code in the patch trys to restore the tape (marked as V4.0) with the wrong out-of-order count! We have used a single version of BRU to create an incompatible, un-restoreable tape!

It should be noted that this patch will not demonstrate this performance if the GBLPAT above does not change the value of OLDPAR from that specified in the original MOV #40.,R2 instruction. But then the patch does not correct the V3.2/V4.0 incompatibility either.

From this authors viewpoint, it is better to cure the known algorithmic errors in LBNORD than to try to treat symptoms of the problems, thereby producing a variety of tape formats that, taken in the extreme, become un-restorable by any version of BRU. It is in this spirit that I present my work on the shortcomings of the distributed version of LBNORD. Use of the LBNORD patch presented earlier should produce a BRU that will create error free, restorable tapes for archival purposes. Older existing tapes (both V3.2 and V4.0) could still be supported by maintaining older versions of BRU (under different names) strictly for restoring archival tapes.

The following is a copy of the SPR response from Digital.

> Thank you for bringing this problem to our attention. We agree with your evaluation of the reported problem and appreciate your help in finding the solution.
>
> We believe that the patch which you submitted with the SPR is correct, but incomplete. It fixes only a part of the real problem and therefore may create additional problems for those who decide to apply it. It corrects the problem with the corrupted quicksort internal stack that resulted in a memory protect violation trap. The corrupted stack might also cause the implemented sort algorithm to work incorrectly, which means that in some cases the output table of retrieval pointers is not sorted correctly. When we decrement $QSTAK+100.(R5) instead of $QSTAK+200.(R5) we definitely corrupt the stack. That may happen when the stack length is more than 50 and is therefore large enough to contain data in $QSTAK+100.(R5). It is not too bad when $QSTAK+200.(R5) is greater by 1 than it should be, but it is much worse that $STACK+100.(R5) is less by 1 than it should be. The size of the upper boundary of an unsorted partition will be incorrectly stated and one element will be missing when the sort algorithm comes to this level of the internal stack.

Now we come to the most important point. According to the quicksort algorithm in Knuth, Volume 3, Sorting and Searching, the stack length should not be more than log base two of the table length and, therefore, in our case, must not be more than 20 words. That means that the problem would never appear if the algorithm was implemented correctly. The upper boundary of the partition in the stack would simply be greater by one and therefore it may cause the stack to be slightly bigger than it should be. However, this size of the stack would never exceed the allocated stack length of 100 words.

The real problem in the implementation is the code that determines which partition should be saved in the stack. LBNORD always saves partitions with the lower values of subscripts and continues processing partitions with the higher subscripts. That is a real problem. We always have to save a larger partition. Only in this case can we be guaranteed that the stack would never exceed the theoretical value. The correction may affect lines 63-73 after the label C00274 in your listing.

Now we are pretty much prepared for the worst. Because the algorithm does not always give us the sorted table, we cannot fix the problem without causing a danger of incompatibility when trying to restore a tape created with old LBNORD which had errors using the new version of this module.

Fortunately, it does not represent a dangerous problem to our users. Even in case the sort algorithm does not work correctly, BRU creates a correct backup set. That backup set can be restored without any problems because the same sort routine works during the backup and restore. It means that all tapes that were created by BRU using this sort algorithm are absolutely restorable. The only problem is when the user experiences a stack overflow fatal error message or a memory protect violation, and, therefore, cannot complete the backup operation. It happens very seldom in very specific cases which we were not able to reproduce on any of our systems. Then the user needs some work around for the problem.

We would greatly appreciate any media or even a printed dump of retrieval pointers table that reproduces the problem.

The general solution for this problem is to change the tape structure by writing a special flag indicating that the tape was created with the new version of sort. During the restore we can check this flag. We will use the old sort procedure if the tape was created before we fixed the errors, and the corrected sort procedure if the tape was created after we fixed the errors. We are also considering replacing the current sort procedure with the new one which will be more efficient. It will require a considerable amount of work. An article with all these corrections will be published in a forthcoming Software Dispatch.

We apologize for any inconvenience in your work caused by this BRU problem. There are several ways to get around this problem. Before you get the corrections you could use DSC in case you encounter a stack overflow problem and switch back to BRU for the next backup. Another solution is to use the procedure described in the Article 5.1.17.9, published in the AUGUST 1980 Software Dispatch.

RSX-11M V4.0 SYSGEN

Greg Liverman

MPI fuer Quantenoptik
8046 Garching bei Muenchen
West Germany

The following are some random observations that I have made during and after my first V4.0 SYSGEN. The incoherence and sketchiness are due to the fact that I am firing this off immediately after doing the SYSGEN so that it can get into print as soon as possible and thereby help the most people. So, enough first paragraph blabber...

I have a RK05 kit. Nine 600' DSC tapes for the system, one 600' RMS-11 tape, one 600' standalone DSC tape and one 2400' Autopatch A tape are in this kit. The first thing to do with it is DSC all those system tapes onto RK05's so that you can use them. I also did something else that will make life a little easier for me in the future. I copied all of those RK05's onto one 2400' tape using BRU64K, the standalone BRU system. Each RK05 is now a BACK_UP_SET on the tape. In the future I can use BRU to get single files, a complete disk or all disks off of the tape. And I can do this on a machine without RK05 hardware. If you have to support a lot of "copy only" licenses like we do, then you can easily see the advantages of this tape. The detailed directory listings of the various kits in Appendix F of the RSX-11M System Generation and Installation Guide (RSX-11M SG&IG) are a big help also. If you create a kit yourself, don't forget to edit SYSGEN.CMD on the new kit and change the kit type symbols. They are located near the beginning of SYSGEN.CMD and are easy to identify.

I performed the SYSGEN on a 11/34 with 124kW of memory and dual RK05's. The target system was a 11/23, 124kW memory, 8 RL02's on two controllers, 4 RX02's on two controllers and 8 terminals on DL11's. I also chose LP, NL and CO drivers and every option that I could get my hands on - FCSRES, full duplex terminal driver with all the bells and whistles, executive commons, and all the executive directives. This means that I did not use autoconfigure or the "default" system.

The V4.0 SYSGEN reminded me very much of a V3.2 SYSGEN. It took me about the same amount of time, the questions were exactly the same in many instances, and it required about the same amount of disk space. I encountered no problems with the RK05 kit. You really only need 5 disks for a complete SYSGEN through Phase III and only 4 if you are only doing a SYSGEN through Phase II. This was the same for V3.2.

A system generation still takes a relatively long time. My times were approximately the same as those for the example SYSGEN in the V4.0 RSX-11M SG&IG. As far as I am able to determine, DEC used an 11/34 with 124kW of memory and dual RL01's for their SYSGEN. They were doing a SYSGEN for the same machine with support for two RK05's on one controller, two RL01's on one controller, a magtape, lineprinter, one DL11 and one DZ11. The table below gives the actual breakdown in terms of elapsed time for various steps:

	PHASE I	
	Me	DEC
Executive ASM	24:14	28:19
MCR Subset ASM	4:36	5:49
Device Drivers	11:33	15:47
Total	40:23	49:55
Elapsed Phase I	1:21:18	1:19:17

	PHASE II	
	Me	DEC
Library Build	9:16	10:51
Exec. Build	4:37	4:41
TTDRV Build	1:00	1:03
Other Drivers	2:21	2:20
FCSRES Build	2:26	2:23
Priv. Utilities	1:26:54	1:23:33
VMR	5:55	3:58
Total	1:52:29	1:48:49
Elapsed Phase II	1:56:40	1:51:10

I am not about to claim that these times are any sort of hard and fast data. However, one can draw the following conclusions:

1. You are going to spend at least 40 minutes answering questions in Phase I if you don't take the "default system" or "autoconfigure" short cuts. I, and I assume DEC also, did not spend a lot of time thinking about the answers to the questions in Phase I, so our times are on the fast side.

2. The driver assembly and task build times depend on how many drivers you are building. The driver assembly times also depend on how many controllers and units you apecify.

3. The executive and MCR assemblies and task builds won't take longer than the times listed above if you have executive commons. Without executive commons, those times might be larger.

4. The Phase II time, as usual, is dominated by the privileged utility task builds. They take a long time even with the FCSRES. That's probably because Phase II creates the CMD and ODL files for these builds dynamically. You can get SYSGEN to save these CMD and ODL files for you. If you do, then a rebuild of these tasks outside of SYSGEN (using FCSRES) should take considerably less time than is indicated here. If you decided against FCSRES, then this step will take even longer than indicated here.

5. Since both DEC and I selected ANSLIB and multi-user MCR, the library build time should be considered an upper limit.

6. The VMR time is using the SYSVMR.CMD provided by SYSGEN. If you edit that file, then your VMR time will probably be different.

My total elapsed time was about 3 hours and 20 minutes from booting the V4.0 disk to the time the VMR was finished. I did it on a nice, quiet Sunday afternoon when there were no phone calls or questions about how one loads a tape on the magtape drive. That's about the same amount of time that I would need for a V3.2 SYSGEN. If you have never done a RSX-11M SYSGEN, then I reccommend trying your first one on a nice, rainy weekend.

I didn't do this first SYSGEN completely correctly on the first try. DEC has changed the way that you specify serial line interfaces. I answered the way I had with V3.2 and didn't discover my mistake until Phase I was almost complete. So I had to start that over. The time wasted on this error is not included in the times above. The RSX-11M/RSX-11S Release Notes describe this change, along with a couple of other similar traps, in section 3.3, page 30.

The Release Notes, section 3.10, page 36, claim RSX-11M is fully transportable among 22-bit machines. All you have to do is to boot and save the virgin system on the machine specified during SYSGEN. I have not tried this, but I think that it should work. The situation is also fairly easy to explain. There are two types of 22-bit machines - UNIBUS and Q-BUS. The UNIBUS '11s ('70, '44, '24) use a UNIBUS map to enable NPR devices to do DMA. The Q-BUS is a 22-bit bus and therefore the NPR devices are expected to do their own 22-bit DMA. DEC has only one NPR Q-BUS device capable of 22-bit DMA - the RL01/RL02 controller (the new one). There is a bit (DV.EXT=400) in U.CW1 of the DL UCBs that indicates if the controller is capable of 22-bit DMA or not. The DL

driver, and only the DL driver as far as I know, checks this bit dynamically to see if the hardware does 22-bit DMA or not. This bit is set or cleared when the UCB is built (by SYSGEN) depending on whether or not the CPU is an 11/23. If it is an 11/23, then the hardware must be able to do 22-bit DMA. If not, then there must be UNIBUS mapping hardware. Therefore, the first boot and SAVe must be done on the target hardware specified at SYSGEN. The next boot could be done on any 22-bit machine because SAV will check the hardware and set the DV.EXT bit in U.CW1 properly.

RSX-11M Sysgen on RK05 System

Dr. Meisenburg

Heinrich-Hertz-Institut
Einsteinfer 37
D-1000 Berlin 10

Our system is a PDP-11/34 with one RK05F and one RK05J disk drive, 128 KW MOS memory, a TS03 magtape, and several terminals. A colleague and me sysgened RSX-11M V4.0 from the RK05 distribution kit for several target systems with different configurations (RK05, RK07, RM02, FPP...).

The first problem occured when DSC failed to copy the RSXM32 tape to disk. After having got a new copy of the tape, the disk copy procedure worked without further problems.

SYSGEN Phase I worked very smoothly, except problems with mistyping some CSR addresses. But that is a problem of the operator rather than a problem of the SYSGEN procedure. Since we have a mapped system, all unneccessary files for unmapped systems were automatically deleted by SYSGEN. We did not use autoconfigure or the standard function feature. We did include virtually all system directives, all terminal handler features, middle FCP, PMD, RMD, DCL, one user-written CLI, user written drivers, error logging, queue manager and queue print spooler, and PMT. We selected the large (20K) executive since otherwise pool space becomes too small.

To save disk space with the system image, we defined memory size for the target system to be 64 KW. That saves 240 disk blocks of scarce RK05 system disk space since RSX-11M automatically expands to the available memory space when booted.

SYSGEN Phase II was done without any problems. But be aware that you must use a virgin copy of PRVOBJ. Otherwise SYSGEN Phase II will fail since a lot of files are deleted after they are used and no longer needed. You can reduce the size of TTPAR as described in chapter 5.6 of the System Generation Manual (we use exactly 4 KW with 8 terminals) and possibly reduce the size of pool (we actually use 4500 words without pool problems) by editting the SYSVMR file. At the end of SYSGEN Phase II, you may exactly follow the post-system generation guidelines.

So far, no disk space problems exist with SYSGEN from RK05's. However when starting with SYSGEN Phase III to build or rebuild tasks that should link to FCSRES or use ANSI magtape support, you soon get into trouble with space on your system disk. To overcome that problem, you should delete those tasks which will be rebuilt except PIP and TKB - they are used during Phase III. This will not be sufficient in most cases though. So you should do SYSGEN Phase III three to four times, building only a subset of the desired tasks and copying the new versions to another disk. This procedure works sufficiently, but it is not very elegant.

Due to mistyping mentioned above, we did several runs of SYSGEN using the saved answer files. Problems only occured when we did not use virgin copies of the distribution tapes. So keep a copy of your saved answer files on a backup disk.

It is then up to the user to find a meaningful distribution of the tasks to two disks, since they do not fit on one system disk. We copied some of the less used tasks to another disk which is mounted on demand.

To get advantage of the excellent new help files, we selected BREIF HELP support which was placed on LB:, usually equal to SY: on RK05-only systems (122 blocks for DCL.HLP and 35 blocks for MCR.HLP). FULL HELP support was placed on one of the remaining two disks. If somebody wants full help support, he has to assign that disk as LB:. The FULL HELP costs a lot of disk space, but may be very useful for the first couple of weeks when users migrate from MCR to DCL or to the new EDT. To build the help files one drive contains the system disk, the second disk is assigned as LB: and should have at least 2000 free blocks, and the third disk contains HLPSYS. After that, follow the guidelines in chapter 6.2.10 of the SYSGEN manual.

We ran the UETP procedure. We used a copy of the system disk which provides the desired utilities (see chapter 9 in the System Management Guide) and which does not contain any unneccessary files. Follow exactly the procedure desribed in that manual and UETP will run and complete successfully.

In conclusion, SYSGEN Phase I and II have proved to run on a RK05-only system with no problems. SYSGEN Phase III causes minor problems with disk space which can be solved by doing it in three to four steps and copying new versions of the tasks to another disk. It is possible to provide full help support for a transitional period with the penalty of 2000 blocks less user space. The UETP procedure can be successfully run with a specially tailored system disk.

Problems in Bringing RSX-11M V4.0 Online

Roger Jenkins

Wycliffe Bible Translators
19891 Beach Blvd.
Huntington Beach, California 29647

As advertized, RSX version 4.0 was very easy to sysgen. I believe our first attempt worked. Our users and operators experienced many frustrations during the first few weeks on 4.0, but I believe that many of these frustrations were due to the fact that we didn't want to burn all of our bridges back to 3.2

until we were sure that 4.0 was working and we did not want to place the 4.0 system on the same disk as 3.2. As a result, we ran with 4.0 on what was to us a non-standard disk. For a while no one knew where any of the system files were and our conversion was not a clean break with the 3.2 system.

Our general impression of 4.0 is that it has a lot of nice features, but we didn't get the increase in pool space that we expected (see #9 below) however, we do not plan to go back to 3.2.

The following comprises a list of the problems we had in bringing RSX version 4.0 on line.

1. We received an RK06/7 kit instead of a Big Disk kit. By playing with DSC and BRU we were able to get both tapes onto one disk. TSC said that it still wouldn't work right, and that I would have to make a lot of changes to the command file to make it work like a Big Disk kit. As it turns out, all that needs to be done is to edit SYSGEN.CMD and change the line

   .SETT $RK06

   to

   .SETF $RK06

   and you have a Big Disk sysgen command file.

2. Due to the change in FCSRES, we had to retaskbuild all of our programs that use FCSRES. This was time consuming, but all of the tasks seem to run ok with the new FCSRES. There was only one program that would not taskbuild with FCSRES. We did not research the problem, but we were able to rebuild the program with the disk resident library, and all went well.

3. The command string parsing subroutines (or at least the FORTRAN interface to them) work differently. A program that used the high level language interfaces to the command string parsing subroutines from the DECUS tapes stopped working. It was determined that previously the program would use the current UIC if no UIC was specified, under 4.0 it would use [000,000] for the UIC. The solution was to remove the following call to CSIDEF:

   ```
   CALL CSIDEF('O','SY:',0,0,'SP ')
   ```

   This required the users of the program to specify "SY:" on all file specifications.

4. Privileged tasks, of course, had to be rebuilt.

5. RMS or the interfaces to it have changed. We were not able to rebuild RMS immediately after installing 4.0. As a result, we could not taskbuild RMS programs. When they were retaskbuilt using the 4.0 SYSLIB, they had many undefined references. Our solution, until we were able to rebuild RMS for 4.0 was to retaskbuild all of these programs using the 3.2 SYSLIB.

6. TDX provides a pseudo command "TYP" which is spawned as a PIP TI:=file. This would not be a problem except that we have our own task TYP which performs differently than the PIP command. Our temporary solution was to install TYP permanantly. Our final solution was to change the TDX command from TYP to LIS(t).

7. TDX prints an extra prompt (>) after each task runs which was run via a flying install. This is mainly noticed when these programs are run from a command file with QUIET ENABLED.

8. The Indirect Command File Processor command, .TESTFILE doesn't work the same. We noticed the problem when testing for the existence of an indexed file. Instead of returning status of 1 indicating that the indexed file exists, the status returned is 325 (Bad record type). I called the telephone support center and they reported that this is not a bug but a feature(!). The new .TESTFILE directive performs a sequential file OPEN. This will cause .TESTFILE to get an error if the file is not a sequential file.

9. Pool seems to be tighter under 4.0. We were running 3.2 with Ralph Stamerjohn's 24k Exec, and it seems that Ralph's mods gave more pool space than 4.0 does. We are really at a loss to completly explain the difference. We have DCL in, but plan to remove it to gain pool space.

10. Command lines that are exactly 80 bytes long (particularly with SORT) don't seem to work. Our solution was to shorten these command lines.

11. The Queue Manager works a little different. We had to modify all of our command files that manage the Queue Manager. This was mainly due to the lower case attribute associated with each printer. (I am afraid that I didn't document the other problems, and now I can't remember what they were.)

    We also had a problem with the Queue Manager due to the changed default in the number of flag pages. 3.2's default was /FL. 4.0's default was /-FL. The result was that users who expected a file flag page on their print jobs didn't get one. We solved the problem by retaskbuilding QUE with a changed default (taskbuilder option).

    The "QUE jobname/MODIFY/LOWER" command doesn't modify the queue entry and doesn't display any error message.

RSX-11M V4.0 SYSGEN on a VAX

Frank J. Nagy

Fermi National Accelerator Laboratory
P.O. Box 500, MS: 306
Batavia, Illinois 60510

We have successfully completed a SYSGEN for RSX-11M V4.0 using the MCR CLI under VAX/VMS. For the most part, the instructions in the RSX SYSGEN Manual appropriate to online SYSGEN's are applicable to VAX/VMS. However, some preparatory work must be undertaken to successfully perform the SYSGEN. Some of this preparation is similar to that needed to do a V4.0 SYSGEN under RSX-11M V3.2 and some is unique to the VAX. One note, V2.5 of VMS was used for this SYSGEN, the consequences of VMS V3.0 relative to an RSX-11M V4.0 SYSGEN are not known at this time.

Our RSX distribution is the RL01/RL02 distribution kit on 9-track magnetic tape. The VAX is equipped with RL02 disk drives which will be used for the SYSGEN itself. However, the version of DSC on the VAX (DSC1.EXE) for Structure Level 1 (ODS-1) disks is compatible with the V3.2 DSC and not the V4.0 DSC. Thus the patch described in the RSX-11M V4.0 Release Notes must be used to create a new version of DSC for the VAX. The DSC library and build files from an RSX V3.2 distribution kit were patched and task built on the VAX with the compatibility mode SYSLIB.OLB in SYS$LIBRARY which includes some patches to allow to the command line parsing and file parsing routines to handle VAX logical names and named directories. The resulting DSC task image was placed in SYS$SYSTEM as DSC1V4.EXE. This program was then used to copy the distribution kit from tape onto RL02 disks. The following DCL command sequence was used to copy the baseline distribution disk onto tape:

```
$ MOUNT/NOWRITE/FOREIGN mt0:
$ MOUNT/FOREIGN dl0:
$ MCR DSC1V4 dl0:/VE=mt0:
```

This sequence was repeated to copy the RSX V4.0 distribution kit onto five RL02 disks.

Before attempting a SYSGEN, several RSX V4 utility tasks were copied from the baseline system disk onto the VAX system disk (i.e., [1,54]MAC.TSK was copied to SYS$SYSTEM:MAC4.TSK). A series of ASN statements (under the MCR CLI) were then placed in the pre-SYSGEN command procedure:

```
$ ASN SYS$SYSTEM:MAC4.TSK=MAC
```

These assignment statements cause the overlays for the tasks to be loaded much faster as the task images on the VAX RM80 disks are used rather than those on the RL02 baseline disk. A listing of the pre-SYSGEN command procedure (RSXSYSGEN.CMD) is included with this note.

Our initial attempts to perform the RSX V4.0 SYSGEN ran into only one problem. This problem is due to an error in the VAX MCR will performs the .EXIT indirect command as if it were a .STOP and exits all command procedure levels. Note that this was under VMS V2.5; this error may have fixed in VMS V3.0. Attached to this note is a listing of the SLP correction file to replace the .EXIT commands in SYSGEN3.CMD with the following two line sequence:

```
IF <SYSTEM> NE 5 .EXIT 1

EXIT 1
```

which will use .EXIT if not VAX/VMS and the EXIT command if on a VAX. In addition to the corrections to SYSGEN3.CMD, similar corrections must be made to all the tttBLD.BLD files in [1,20] on the EXCPRV and RLUTIL disks. These command procedures to construct the .CMD and .ODL files for the privileged and non-privileged tasks use the .EXIT directives in (usually) two places as ".EXIT 1" and ".EXIT 4". The same solution applied to the SYSGEN3.CMD file (see above) can be applied to the .BLD files.

Before the RSX SYSGEN can begin, the VMS system parameter (CLISYMTBL) controlling the size (in pages) of each process's CLI symbol table must be modified. The normal setting for CLISYMTBL is 20; a value of 40 was used for doing the RSX SYSGEN. The parameter is modified by logging in as a privileged user and running the VMS SYSGEN program. This operation requires the CMKRNL privilege. The commands used to modify CLISYMTBL are thus:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>USE ACTIVE
SYSGEN>SET CLISYMTBL 40
SYSGEN>WRITE ACTIVE
SYSGEN>EXIT
```

One must then logout since to do the RSX SYSGEN you must have MCR as your Command Line Interpreter. In addition, the CLISYMTBL parameter only takes effect when a process is created (i.e., login time).

To actually do the SYSGEN, login again using the "/CLI=MCR" qualifier on the user name. This will establish MCR as the default command line interpreter (CLI) allowing the processing of RSX indirect command files. The RSX V4.0 baseline system is mounted, the UIC is set to [200,200] and the first SYSGEN command file is executed:

```
> MOU/SHARE d10:RSXM32
> SET /UIC=[200,200]
> @SYSGEN
```

All phases (I, II, and III) of SYSGEN are then performed as they would online to an RSX system. The RSX Autoconfigure program will not be run as is to be expected. The final step of booting and saving the newly SYSGEN'ed system must be performed on the target PDP-11 system as must the test run of UETP used to checkout the newly created system.

One note of caution, since our PDP-11's do not include any of the K-series peripherals, we have never done an RSX SYSGEN with those features. Thus we do not know if those portions of the RSX SYSGEN will be performed correctly under VAX/VMS.

Listing of file RSXSYSGEN.CMD

```
.;
.; RSXSYSGEN.CMD
.;
.; Used to make the logical name assignments to do an RSX SYSGEN
.; on the VAX under VMS.  From the VAX-11/RSX-11M USER''S GUIDE
.;
.; Modification History:
.;
.;  01-Jul-81    FJN     Make sure MAC used from SYS$SYSTEM:
.;  03-Jul-81    FJN     Straighten out assignments and
.;                       make sure utilities used from SYS$SYSTEM
.;  05-Jul-81    FJN     Use BIGTKB.TSK on SYS$SYSTEM:
.;  25-Jan-82    ADT/KJC Modify message concerning spooled maps
.;                       and the MOU/SHARE command for clarity.
.;  08-Jul-82    FJN     Ask question re RSX V4.0 SYSGEN to use
.;                       xxx4.TSK utilities on VAX system disk
.;                       instead of the V3.2 forms.
.;
.ENABLE SUBSTITUTION
.;
.; Get RSX target device if not entered as a parameter
.;
.IF P1 EQ "" .ASKS P1 Target device name
.;
.; Make sure the device name is ended by a colon
.;
.SETS TEMP P1[<STRLEN>:<STRLEN>]
.IF TEMP NE ":" .SETS P1 P1+":"
SHOW SYMBOL P1
.;
.; Set default to target device
.;
SET DEFAULT 'P1'
```



```
.;
.; Assign SY, SYO, LB, and LBO to target disk
.;
ASN 'P1'=SY:
ASN SY:=SYO:
ASN 'P1'=LB:
ASN LB:=LBO:
.;
.; Equates MPO and MP, also TKO and TK
.;
ASN MP:=MPO:
ASN TK:=TKO:
.;
.; If doing RSX V4.0 SYSGEN, use different assignments
.; for the utilities.
.;
.ASK V4 Doing an RSX-11M V4.0 SYSGEN
.IFT V4 .GOTO RSX4
.;
.; Assigns equivalence names for TKB and VMR
.; BIGTKB.TSK was directly copied from an RSX system.
.;
ASN SYS$SYSTEM:BIGTKB.TSK=TKB
ASN SYS$SYSTEM:VMR=VMR
.;
.; Make sure utilities used from SYS$SYSTEM:
.;
ASN SYS$SYSTEM:MAC=MAC
ASN SYS$SYSTEM:EDT=EDT
ASN SYS$SYSTEM:LBR=LBR
ASN SYS$SYSTEM:PIP=PIP
.;
.; Set process privileges needed to do a SYSGEN
.;
SET PROCESS/PRIVILEGES=(SYSPRV,LOG_IO,CMKRNL)
;
; Note that the baseline system TKB is used.  The maps
; cannot be sent to a non-RSX device or to magtape.  They
; may be put on the target disk ('P1').  It must be mounted
; SHARE'd to allow the maps to be spooled to the printer:
;
;       > MOU/SHARE 'P1'RSXM26
;       > SET /UIC=[200,200]
;       > @SYSGEN
;
EXIT
.;
.; What follows is specific for V4.0 gens
.;
.RSX4:
.;
.; Assigns equivalence names for utilities kept on VAX
.; system disk (to allow speedier overlays).  Note that
.; these were copied/built from RSX baseline system or
.; RSX system libraries and cannot reference VAX/VMS device
.; and directory names.
.;
```

```
ASN SYS$SYSTEM:TKB4.TSK=TKB
ASN SYS$SYSTEM:VMR4.TSK=VMR
ASN SYS$SYSTEM:MAC4.TSK=MAC
ASN SYS$SYSTEM:LBR4.TSK=LBR
ASN SYS$SYSTEM:PIP4.TSK=PIP
.;
.; Use EDT from VAX/VMS system disk (native mode)
.;
ASN SYS$SYSTEM:EDT=EDT
.;
.; Set process privileges needed to do a SYSGEN
.;
SET PROCESS/PRIVILEGES=(SYSPRV,LOG_IO,CMKRNL)
;
; Note that the baseline system MAC and TKB are used.
; The listings and maps cannot be sent to a non-RSX
; device or to magtape.  The maps may be put on the
; target disk ('P1').  The assembly listings may be
; placed on a scratch disk (RL01/RL02).  Each disk
; must be mounted SHARE'd to allow the printouts to
; be spooled to the printer:
;
;       > MOU/SHARE DL2:LISTINGS
;       > MOU/SHARE 'P1'RSXM32
;       > SET /UIC=[200,200]
;       > @SYSGEN
;
EXIT
```

Listing of SYSGEN3.CMD corrections file

```
SYSGEN3.CMD;2/-AU=SYSGEN3.CMD;1
-23
 .;
 .; Modified:
 .;
 .;  04-Jul-82   Frank J. Nagy   Fermilab Accelerator/Controls
 .;               Use EXIT instead of .EXIT for VMS V2.5
 .;
-818,818
 .;
 .; For VAX/VMS V2.5 use EXIT rather than .EXIT (as .EXIT
 .; incorrectly behaves like .STOP and exits all command procedures!).
 .;
 .IF <SYSTEM> NE 5 .EXIT 1
 EXIT 1
-822,822
/
```


Using the RSW-11M V4.0 .BLD Files

Frank J. Nagy

Fermi National Accelerator Laboratory
P.O. Box 500, MS: 306
Batavia, Illinois 60510

Correction from Frank Nagy

I have just discovered two minor errors in this article. The names of the symbols SYPAR and FPAR, defined to be strings for TKB "PAR=" option, should be $SYPAR and $FPAR.

In addition, I have a bit of information for people doing RSX development on the VAX. The RSX utilities distributed with VMS V3.0 are those for RSX-11M V4.0. This was pointed out to me by Telephone Support as being mentioned in the VAX-11/RSX-11M User's Guide on page vii. In addition, TSC verified that the VMS TKB (under VMS V3.0) was the same version as the RSX-11M V4.0 TKB.

We had occasion to need .CMD and .ODL files for RSX-11M V4.0 utility tasks in a situation where it was impractical to use SYSGEN3.CMD to construct those files. In particular, this came about because we had no SGNPARM.CMD file available from a previous (non-existant) SYSGEN. A small amount of effort was invested to determine how the .BLD files interact with the SYSGEN3 command file. The results of that investigation are reported here for the use of the RSX community.

The .BLD files for the RSX-11M V4.0 privileged and non-privileged utilities are indirect command files which create specific instances of the .CMD and .ODL files for the utility tasks. The .CMD file for a task is then used as input to the task builder to produce an executeable task image (the .ODL file contains the Overlay Description Language text for those utilities which use overlays). The .BLD files particularize the .CMD and .ODL files through the use of global symbols defined in the outer indirect command file which calls the .BLD file. Normally this outer command file is SYSGEN3.CMD, but a user can write his own indirect command files which define the global symbols and call the .BLD files to construct particular instances of the .CMD and .ODL files for RSX utility tasks. Conversely, a user supporting multiple systems should be able to construct .BLD files for his own tasks and have SYSGEN3 particularize user tasks in the same manner as RSX utility tasks are handled.

The information about the meanings of the global symbols used in the .BLD files has been gleaned from a study of the SYSGEN3.CMD indirect command file and several of the .BLD files. This information is not complete as there is some doubt about the interpretation of some of the symbols. The examination of SYSGEN3.CMD and the .BLD files was not exhaustive, but the information gained and presented here will probably be sufficient for most cases. Possibly someone else can shed some light on the remaining items.

1.0 Device Specification Symbols

SYSGEN3 and the .BLD files use a set of symbols whose values define the input and output devices for the operation of the .BLD files and the task builder. These symbols specify both a device name and a directory (UIC). The string symbols and their uses are:

$BL Location of .BLD file (used in SYSGEN3 to invoke the .BLD indirect command file).

$CL Location of .CMD and .ODL files created by executing the .BLD file (also used in SYSGEN3 as input location of .CMD file for the task builder).

$TK Output location for task image (.TSK) created by the task builder. Also used as the location of the executive symbol table file (RSX11M.STB) in the privileged task .CMD files.

$MP Output location for task builder maps (.MAP).

$OD Used in the .CMD files as the input location of the .ODL files.

$LI Use in the .CMD and .ODL files as the input location of the object module library (.OLB) of the task being built.

Two string symbols are used to modify the filenames of the files created by the .BLD command file and the subsequent task building. The $TYP1 symbol is used in the .BLD file to modify the file names of the created .CMD and .ODL files. Normally these file names have the form "xxxBLD" where "xxx" is the 3-letter name of the utility task built by the files (note that the indirect command file used to make the .CMD and .ODL files has a file specification of the form "xxxBLD.BLD"). The value of the $TYP1 symbol provides 0 or 3 characters to be inserted into the file names between the task name, "xxx", and the "BLD" (as in PIPRESBLD). The values of the $TYP1 symbol depend upon several additional conditions:

ANS for RSX-11M tasks built with ANSLIB,
RES for RSX-11M/M-PLUS tasks built with FSCRES, or
FSL for RSX-11M-PLUS tasks built with FCSFSL.

If none of these conditions are true (i.e., RSX-11M tasks built with SYSLIB), then the value of $TYP1 is a null string. A second symbol, $TYP2 is used as a modifier for the file names of the .TSK, .MAP, and .STB files produced by the task builder. This symbol will normally only have a null string as its value for RSX-11M and either "RES" or "FSL" for RSX-11M-PLUS (or the null string). SYSGEN3.CMD can assign either "ANS" or "RES" to $TYP2 for RSX-11M ($11M is TRUE) if $SYBLD is also TRUE. A comment indicates this is only TRUE for SYSBLD (possibly this has to do with IAS). Some of the .BLD indirect command files ignore the value of $TYP1. In such a case these command files also set the symbol $UTYP1 to FALSE to indicate that $TYP1 has not been used (this is required for SYSGEN3).

Some of the standard RSX-11M/M-PLUS utility tasks have names longer than 3 letters and are shortened as shown:

ACN	ACNT
LPI	LPINIT
MCM	MCRMU
MTA	MTAACP
QCL	QMGCLI
QPR	QMGPRT
SHU	SHUTUP
TKN	TKTN

This table was extracted from SYSGEN3.CMD.

2.0 Target System Specification Symbols

Another set of symbols is defined by SYSGEN3 from the SGNPARM file to identify the characteristics of the system for which the utility task is being built. Several of these symbols identify the type of system and other characteristics:

$11M Set TRUE if target system is RSX-11M.

$11MPL Set TRUE if target system is RSX-11M-PLUS.

$IAS Set TRUE if target system is IAS.

$MAP Set TRUE if target system is mapped (can be FALSE only for an RSX-11M system).

$MAPM Same as $MAP.

$MMSW Task image switch for TKB. Has value of "/MM" for mapped target systems ($MAP is TRUE) and "/-MM" for unmapped target systems ($MAP is FALSE). For IAS this symbol is a null string.

$MUL Set TRUE if multiuser task images are to be made (only for RSX-11M-PLUS or IAS systems).

$MUSW Task image switch for TKB. Has value of "/MU" if a multiuser task image is to be made ($MUL is TRUE) and "/-MU" otherwise (null string for RSX-11M).

Several symbols are defined with string values and used by the .BLD command files to generate a "PAR=" task builder option line. Which symbol is used is dependent upon which task is to be built. The names and values of the various symbols are listed for both mapped and unmapped systems ("base" represent the base address of the partition for an unmapped system):

	Mapped Systems	Unmapped Systems
$GEN	GEN:0:0	GEN:base:40000
$GENX	GEN:0:40000	GEN:base:40000
$GENB	GEN:0:70000	PAR14K:base:70000
SYPAR	SYSPAR:0:12000	SYSPAR:130000:12000
$PRPAR	GEN:0:0	SPLPAR:130000:10400
FPAR	FCPPAR:0:size	not defined

The FCPPAR size is set to "24000" if FCPMDL (middle) was selected and to "44000" if FCPLRG (large) was selected in SYSGEN. Note that the values listed above are those assigned by SYSGEN3.CMD.

3.0 Library Specification Symbols

Another set of symbols select the resident shared library with the the utility tasks are to be linked (if any) and the default system library to be searched. The names and values of these symbols are:

$FRLTK Set to TRUE if the FCSRES shared library is to be linked to the task (SYSRES for IAS). Otherwise this symbol is FALSE.

$FSLTK Set to TRUE if the FCSFSL (Supervisor mode FCS resident library) is to be linked to the task (only for RSX-11M-PLUS). For IAS and RSX-11M, this symbol is FALSE.

$FCSTK Set to TRUE if neither $FRLTK or $FSLTK are TRUE.

$EXRTK Seems to be the same as $FCSTK as far as can be determined.

$ANFCS Set to TRUE if ANSLIB should be searched instead of SYSLIB (normal case with $ANFCS set to FALSE).

$ANSLB Same as $ANFCS.

$SYGRP Set to the group number of the UIC in which the libraries and symbol table files of resident libraries are stored. Set to "1" for RSX-11M/M-PLUS to use [1,1].

$LIBOP If a shared resident library is used, then this symbol is set to the string specifing the task builder option to link the resident library. If a resident library is not being used, then this is set to the null string. The values of this symbol are:

for FCSRES(11M/M+): "RESLIB=[1,1]FCSRES/RO",
for FCSFSL(M+ only): "SUPLIB=FCSFSL:SV", and
for IAS: "LIBR=SYSRES:RO".

4.0 Comment Symbols

These symbols are used to modify the comments in the .CMD and .ODL files, constructed by executing the .BLD command files. The string values assigned to these symbols by SYSGEN3.CMD and the context in which they are used is (value is enclose in quotation marks):

$COM Identifies for what system the files are begin made, one of:

"ON AN UNMAPPED RSX-11M SYSTEM",
"ON A MAPPED RSX-11M SYSTEM",
"ON AN RSX-11M -PLUS SYSTEM", or
"ON AN IAS SYSTEM".

$BLDID Identifies who constructed the files: "SYSGEN3.CMD VERSION 1.81"

$DFLIB Identifies the default library: "SYSLIB" or "ANSLIB".

$RSLIB Identifies the resident library linked to, one of: a null string, "FCSRES", "FCSFSL", or "SYSRES".

An Exposition on SAV in RSX-11M-PLUS V2.0

Paul J. Bezeredi

Digital Equipment Corporation
Nashua, New Hampshire

I. Why SAV?

The general intent of SAV is to copy the current system image from main memory back into the system image disk file from which the system was booted. This causes the current state of the running system, including modifications made since the system was booted, to become a permanent part of the system image file. Therefore the modifications do not have to be made whenever the system is booted. For example, tasks do not have to be reinstalled every time the system is booted and patches (made by OPEN or XDT) need be made only once.

SAV, and the indirect command file it invokes, will automatically handle the many details necessary to use any system. For example, redirecting the pseudo devices, mounting the system disk, and utilizing all of available main memory.

There are two other reasons for saving a system. SAV provides the only way a system image can be made hardware bootable, that is, for the disk's hardware boot block, Logical Block Number (LBN) 0, to be written so it points to the system image. Additionally, after being saved, the disk on which the system resides can be compressed (copied by BRU or DSC) without destroying the integrity of the system.

II. The Concepts and Techniques of SAV

SAV operates in two distinct phases. In the first phase, SAV writes what is in main memory into the system image file, and it is executed only by invoking SAV from MCR. In the second phase, SAV restores the saved context to the CPU and it is invoked either automatically in the first phase or by booting the saved system image file.

In phase one, the lowest section of main memory (bytes 0 through LOWSIZ*2) is copied into a buffer in SAV's address space (see the module SAVE for the definition of LOWSIZ). The system image bootstrap (see $VBN1 in the module SAVVB1) is copied into the locations just vacated. The special driver for the type of system disk is copied into the system image bootstrap. (See Section V, "Special Drivers", for details about special drivers.) The mapping registers are pushed onto SAV's stack and a copy of the the special driver that is in SAV's address space writes all of main memory into the system image file.

Basically, phase two reverses that process. Control of the CPU is passed to the system image bootstrap from phase one by jumping to location zero or by using BOO, which reads the system image bootstrap from the front of the system image file into low main memory.

The special driver in the system image bootstrap reads the rest of the system image file into main memory. Control is passed to the task SAV, which overwrites the system image bootstrap with the orginal contents of low main memory, restores the mapping registers from the stack, performs housekeeping functions (for example, expanding or contracting the last partition in the system to include all of available main memory, redirecting the pseudo devices SY:, LB:, and SP: to the booted device, and initiating the startup indirect command file), and exits.

A part of phase one and two deserves special mention. Every installed task has a Task Control Block (TCB) in the system data structures. Offsets T.LBN and T.LBN+1 contain the LBN of the task image file on the disk, so the Executive can quickly locate the file for initial task loading. If the disk is compressed (copied) by BRU or DSC, the task images will probably change locations on the disk, and therefore the system data structures will be wrong. To avoid this, SAV retrieves the task's File ID, which does not change when a disk is compressed, from the task's header and saves the File ID in the task's TCB when the system is written out. SAV converts the File IDs back into LBNs when the system image file is read into main memory.

All active copies of multiuser tasks have a similar problem. Their Partition Control Blocks (PCBs) contain the LBNs of their common read-only sections. When the system image is written out, SAV converts those LBNs to numbers relative to the first block of the task image files that contain the common read-only sections. SAV reconverts the relative block numbers back into absolute block numbers when the system image file is read into main memory.

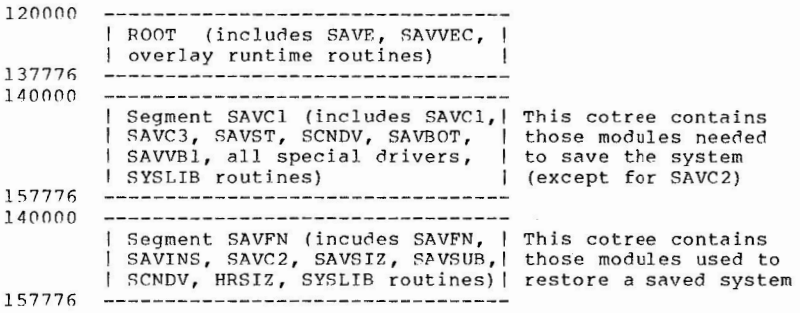
The M-PLUS version of SAV is memory resident overlaid for two reasons. First there is more code in SAV to fit into the 8K words allotted to a privileged task that references the I/O page. The modules SAVC1, SAVC3, SAVC2, SAVST, and SAVFN all used to be part of the module SAVE. There were no subtle reasons for breaking SAVE up into those particular modules. The present arrangement was chosen because it made both memory resident overlays about the same size. Second, because each of

the memory resident overlays is loaded before main memory is copied to the system image file, SAV.TSK can be deleted after the system is saved, and the system will still successfully boot, as long as SAV does not have to be checkpointed in order to mount the system disk.

In the future M-PLUS may support new system disks. Therefore SAV will have to add more special drivers. Such an upgrade should not require reoverlaying SAV (should not increase its virtual address space) but should require modifying the fewest possible modules. Adding another special driver merely requires the writing of a new special driver module and the updating of SAVBLD.ODL.

Because manual overlay loads cannot be mixed with automatic overlay loads, the loading of memory resident code segments such as SAVC1 is also done manually. This is a bit unfortunate, because the implicit loading of code segments is both less confusing and less cluttered. Figure 1 gives a graphic description of the memory layout of SAV.

FIGURE 1 -- MODULE LAYOUT OF SAV.TSK



In M-PLUS, SAV, BOO, and VMR all share a technique for determining the length of a system image bootstrap. SAV symbolically defines the number of disk blocks that compose the bootstrap and derives the values of BTADD and LOWSIZ from this. BOO and VMR get the length of the system image bootstrap minus one from the "transfer address" of the system image file's label block, where SAV leaves it. As a consequence, M-PLUS can boot RSX-11M systems, which have a smaller system image bootstrap, but by coincidence (as far as RSX-11M is concerned) have the appropriate value (zero) as the "transfer address."

In RSX-11M, SAV, BOO, and VMR all locally define the length of a system image bootstrap, BTADD, and LOWSIZ (watch out, all the references may not even be done symbolically). As a consequence, RSX-11M pre-V4.0 systems cannot boot M-PLUS systems, because the 11M BOO has no way of knowing that it must read in more than the one block that is normal for a saved RSX-11M system image file.

The current state of RSX-11M V4.0 is that BOO has been modified to interpret the "transfer address" of a system image file in the same manner as M-PLUS. That is, BOO can boot both RSX-11M and M-PLUS systems. SAV needs to be modified to consolidate the definitions of BTADD and LOWSIZ and to always use the symbolic definitions. VMR (see the modules RANIO and SETUP) needs to be modified to use the "transfer address" from the system image file's label block rather than explicitly knowing how big the system image file bootstrap is.

III. SAV's Restrictions

The restrictions imposed by SAV come in two general categories: those detailed checks on the state of a system that must be passed to guarantee that the system can be successfully saved and the general implications of the current structure of SAV.

For a system to be saved, it must be in a specific state, that is defined in general terms as being "quiet" or inactive. Checks that are made are sensitive to the detailed capabilities of the system (see Section IX, "What SAV does in Detail") and therefore change from base level to base level. When a check fails, you will receive a specific error message.

The manner in which SAV is currently implemented results in several general restrictions:

- If a system is saved on one type of disk (for example an RL02), it generally cannot be copied by BRU or DSC to another type of disk (for example an RK06) and remain bootable. This is because the driver in the system image bootstrap can generally boot only one generic type of device (in the example, this would be an RL01 or RL02). The exception to this are MASSBUS devices (RM02/03/05/RM80, ML11, RP04/05/06/07), RK611/711 devices (RK06 and RK07), and UDA50 devices (RA80). A significant decrease in packaging complexity has been achieved for M-PLUS by combining the MASSBUS, RK611, and UDA50 special drivers into one common driver (in the module SAVCM) and using a controller detection algorithm and the device drivetype code supplied by the hardware to determine the disk type and its associated geometry.

- A saved system cannot be booted if the hardware boot block or the system image boot blocks contain ECC correctable (or any other) errors. This is because neither the ROM boot code, which reads in the hardware bootstrap block, nor the special driver in the hardware bootstrap block, which reads in the system image bootstrap, is large enough to perform ECC correction. They merely retry an unsuccessful operation, which causes an infinite loop, or simply halt.

- Because the M-PLUS system image file bootstrap is more than one block long, its special driver is big enough to handle ECC correctable errors in the system image file. Because the system image file bootstrap for RSX-11M is only one block long, it is too small to hold the ECC correction code. Therefore the RSX-11M boot process is vulnerable to an ECC correctable error anywhere in the system image file.

While an M-PLUS system can boot an RSX-11M system, the inverse is not necessarily true. There are two theoretical reasons for this. First, a saved M-PLUS system requires that CPU registers R4 and R5 contain the physical unit number and CSR address of the boot device. A pre-V4.0 RSX-11M system does not pass this information. (For a complete description of register conventions, see the module SAVVB1.) Second, when reading a system image bootstrap into main memory, a pre-V4.0 RSX-11M system reads in only one block. An M-PLUS system image bootstrap consists of multiple disk blocks.

The attempt to boot a saved M-PLUS system image file with an RSX-11M pre-V4.0 BOO will result in a NO TRANSFER ADDRESS error message before either of the two problems are encountered. This is because the "transfer address" of the saved M-PLUS system is currently 1. The pre-V4.0 BOO mistakes this for the transfer address of a virgin system image file and rejects it because it is odd.

Note the implications of the /CSR switch (see Section VI, "Format of the SAV Command Line"). If the switch is not used, a system can be saved and booted on any controller on the system as long as the hardware bootstrap returns the CSR address of the controller in CPU register R1. To the best of our knowledge, all bootstraps that M-PLUS is likely to run on behave in this fashion.

Debugging SAV is tedious. It is a privileged task that must reference the I/O page, so it is restricted to a maximum size of 8K words. In spite of being overlaid, SAV is too big to allow the inclusion of ODT. The trick of using ZAP to insert BPT instructions in sections of code that run in Kernel Mode (and using XDT as the debugging aid) doesn't always work, because SAV frequently gets into Kernel Mode by jamming the Program Status Word rather than using CALL $SWSTK. The result of this practice is that $STKDP is greater than 0 when the Executive handles the breakpoint and therefore the Executive does not correctly determine how to handle the trap.

There is an additional complication. When exiting XDT, the various Page Descriptor Registers (PDRs) are jammed to be appropriately read/write or read-only. They must all be read/write for parts of SAV. See the module SAVE to identify which sections are restricted in this manner.

The result of this is that infinite loops inserted by ZAP and manipulated from a debugging console are frequently the debugging method of choice.

Many parts of SAV (for example, the memory sizing routine in the module SAVSIZ), manipulate the memory management registers. They obviously must not remap the register that maps their instructions. As a result, such sections of code must be mapped by certain Active Page Registers (APRs). SAVBLD.ODL identifies such sections and the APRs that must be used to map them.

IV. Virgin Systems

The task builder produces virgin (unsaved) system image files. They are characterized by having transfer addresses in their label blocks that point to INITL or XDT. Various fields in the system's data structures, such as T.LBN in each TCB, contain Logical Block Numbers. A saved system image file has a "transfer address" of less than 60 (octal), SAV as the active task, and such words as T.LBN in each TCB containing pointers that are invariant across disk compressions.

Although useful work can be accomplished using a virgin system by manually mounting the system disk, establishing a checkpoint file, bringing devices online, etc., this is not a design goal of M-PLUS. The system may be limited in the future so that the only legitimate operation that can be performed using a virgin system is saving.

A virgin system image will not generally function correctly if its disk has been compressed (copied by BRU or DSC). This is because the Task Control Blocks (TCBs) of the installed tasks and some Partition Control Blocks (PCBs) of checkpointable commons contain absolute pointers to disk LBNs. When a disk is compressed, the starting LBNs of common and task image files generally change. As explained previously, one of the big tricks of SAV is to convert those absolute pointers into pointers that are invariant across disk compressions.

A virgin system image file can be copied by using PIP. If a virgin system image file is on a disk that has been compressed, it can be salvaged by using VMR to remove all tasks and commons and then reinstalling them.

Because a virgin system does not size the system disk, SYSGEN must produce a Unit Control Block (UCB) for the system disk that correctly indicates its geometry (for example, an RK06 or RK07) if the system is to work once it has been booted.

Because of a "trick" that can be used when answering the SYSGEN peripheral questions, there is a bit more flexibility available than is immediately obvious. Currently, for the RL01/02, RP04/05/06, and the RM02/03, the only difference between the large disk of the generic type (an RP06) and its equivalent small disk (an RP04) is the number of cylinders per disk. Therefore when you are given a choice of drive types, choose the largest type. However, note that the RK06/07, RM80/RM05, and RP07 must be dealt with honestly. If the drive turns out to be the larger type, everything is perfect. If the drive turns out to be the smaller, it cannot have LBNs that are too big (LBNs that would push the heads off the end of the disk).

A virgin system does not have to be "quieted" by a CON OFFLINE ALL command before being saved. The I/O data structures output by SYSGEN show all the appropriate controllers and units (everything except the booted disk and console) as offline.

V. Special Drivers

Special drivers are used by SAV to write the contents of memory to the system image file and to read the system image file into main memory when the saved system image file is booted. When SAV writes the hardware boot block, SAV includes a special driver that will read in the system image bootstrap of the saved system image file. The source modules for SAV's special drivers are found in UFD [12,10] and are named SAVdd, where dd is the device name. For example, SAVDL is for the RL01/02 and SAVCM is for the MASSBUS, RK06/07, and UDA50 devices.

BOO also uses special drivers to read system image files into memory. BOO can handle DECtape, the RX01, and the RX02 (see the module BOODRV) in addition to the devices SAV can handle.

A special driver is special because of the constraints under which it must work. A special driver must not use interrupt vectors (while the special driver is being used, the system image bootstrap occupies low main memory). It must be as small as possible (because it must fit into the hardware boot block). It must conform to an elaborate set of conventions established because the driver must be able to read or write variable amounts of main memory or system image files.

The small size (you do not want to waste instructions making tests) and need for flexibility are met by using SAV to modify the driver before inserting it into one of the various environments (hardware boot block, system image file bootstrap, or internally in SAV) the special driver can handle. The places to make the modifications (for example, where to start writing, whether to write or read, etc.) are specified for each special driver by a table.

Structurally, a special driver consists of a module that has three sections:

- The first defines the names of the CSR offsets that the driver uses.
- The second is a fixed length segment of the PSECT DRVTAB that defines various aspects of the driver. For example, its length, what a write function is, and what it's segment name is. In the aggregate, the segments form a contiguous table that is terminated by a zero word.
- The third section is the driver itself.

When the driver is entered, the following CPU registers have special significance:

R5 - will be zero (indicating that the driver should use the CSR address stored in the driver) or nonzero (indicating that the contents of R5 should be used as the CSR address).

R2 - will be zero if no UMRs are required by the special driver else it will be nonzero.

R3 - will contain the BAE offset if R2=0 and the special driver being used is SAVCM or SAVDL else it will be undefined.

Locations 4 and 6 will contain the LBN where the driver should start reading or writing.

When the driver has finished, the following registers will contain information required by SAV:

R0 - will be the residual block count (the number of blocks that were to have been read or written, but were not).

R1 - will be the ASCII representation of the load device name, that is the name of the device that was just used.

R4 - will be the physical unit number of the device that was just used.

R5 - will be the CSR address of the device that was just used.

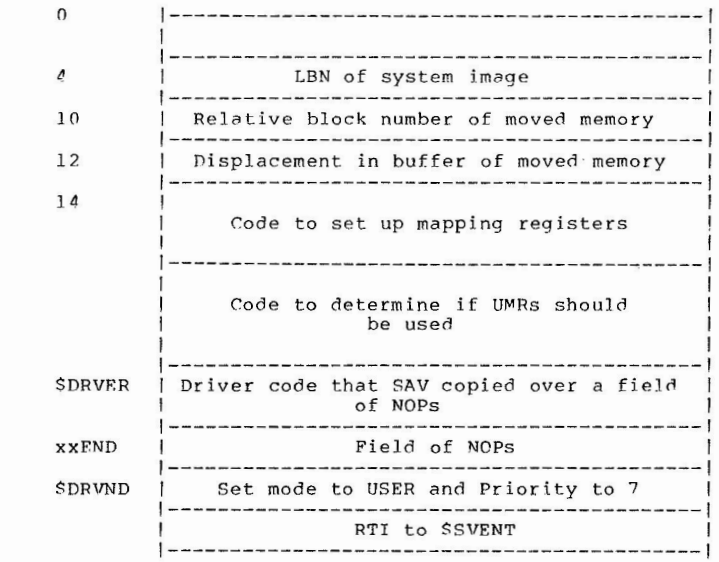
Although the best way to write a special driver is to follow closely one of the existing special drivers, several subtleties deserve mention.

All drivers end with a RETURN (labelled xxEND, where xx is the name of the special driver). When SAV copies the special driver into either the hardware boot block or the system image file bootstrap, it copies up to, but does not include, the xxEND statement. Thus when the driver is finished, control falls into the next instruction of the bootstrap. When SAV uses the special driver to write main memory to the system image file, the special driver is called, so the xxEND statement returns control to SAV. Figure 2 gives a graphic description of how the special driver fits into the system image bootstrap.

Special drivers must be extremely careful about using the stack, because there is not much available. For example, a special driver is about to read in the system image bootstrap into locations 0-2776. The stack is at 3004. If the driver pushes three words onto the stack before starting the transfer, the last word will be overwritten by the system image bootstrap.

Additionally, the use of a "trap catcher" by the special driver can easily push enough onto the stack to extend it into the last few words of the boot block.

FIGURE 2 -- SYSTEM IMAGE BOOTSTRAP (VBN1-3)



VI. Format of the SAV Command Line

The format of the SAV command line is:

SAV[E] [/WB][/MOU="switches"][/SFILE="file spec"][CSR=x]

/WB is an optional switch that indicates that the hardware boot block (LBN 0) of the boot device should be rewritten to point to the system image file that is about to be written.

/MOU="switches" is an optional switch that provides a string to append to the MOUNT command for the system disk. This allows the overriding of the default mount parameters for the system disk. The pair of double quotes is part of the required syntax. SAV simply appends all the characters between the quotes to the MOUNT command. It does no syntax checking.

/SFILE="filespec" is an optional switch that provides the file specification of an MCR indirect command file to use in place of "booted-device:[1,2]STARTUP.CMD", which is the default file specification. The pair of double quotes is part of the required syntax. SAV does no syntax checking on the file specification. It simply precedes the characters between the quotes with an "@" and queues the resulting string to MCR.

/CSR=x is an optional switch that specifies the CSR address for the boot device. If this keyword is not specified, SAV uses the CSR address in the boot device's KRB for writing the system image out to the disk and what the hardware bootstrap leaves in CPU register R1 for reading it in when the system is booted. If x is an even octal number greater than 157777, SAV uses the KRB CSR address to write the system out and x to read it in. If x is "SY", SAV uses the KRB CSR address to both write the system out and to read it in. An implication of this is that given the proper type of hardware bootstrap, the disk can be booted on any controller in the system, not just on the controller on which the system was saved.

This switch is useful for creating distribution kits on systems where the CSR of the actual device is not at the default CSR address for that type of device. For example, on a system with RP05s and RM03s, the CSR address for the RM03 will not be the default, because the RP05s occupy the default address. To create a disk that will boot on a default RM03, save the system with CSR=176700. The system will reboot (the CSR address used to write the system out is taken from the RM03's KRB, and the system image bootstrap that has been left in low main memory is forced to use that CSR address to read the system back in). Use the CON command in VMR to set the vector and CSR addresses for the RM03 to the defaults.

VII. A Roadmap of SAV's Modules and Entry Points

Module	Entry Point	Function
SAVE	SAVEP	Load overlays and call subroutines, push mapping registers onto stack, copy low main memory into a buffer, put the system image bootstrap in low main memory, and transfer control to it
	$SVENT	Restore low main memory, hardware map registers, and CPU context
	$KRBOF	Put KRB offline and remove vector
SAVC1	$SAVC1	Parse command line, unstop tasks, tell PMT to exit
SAVC3	$SAVC3	Check memory size, checkpoint all R/W commons, insure checkpoint files are inactive, check active processors
SAVC2	$SAVC2	Dismount load device, check for tasks with outstanding I/O, insure all devices are offline
SAVST	$SAVST	Set up special driver, check home block, and write label block

	$SAVDN	Setup hardware boot block, set TI: and SY: offline
SAVINS	$STFID	Convert T.LBNs and P.LBNs
	$INSTK	Restore the T.LBNs and P.LBNs
SAVVB1	$VBN1	Initialize the CPU, read system image file into main memory, and transfer control to it
SAVSIZ	$STCLK	Select clock and initialize clock and Floating Point Processor vectors
	$TSTPY	Set up nonexistent CSR table
	$CRSIZ	Size main memory
	$STCPU	Determine CPU type
HRSIZ	$SDISK	Size the booted device
SAVFN	$SAVFN	Finish bringing the system up
	$SAVID	Output console ID, redirect pseudo devices

VIII. What SAV does in Detail (Saving the System)

The following is basically a nondefinitive, English version of SAV's code (see modules SAVE, SAVC1, SAVC3, SAVC2, SAVST, SAVFN, SAVVB1, SAVSIZ, SAVINS, and SAVSUB). It is intended to give a flavor of that code and some of its motivation. The code, not this, is the truth.

The principal reason for the order of the following tests is for coding convenience or for historically obscured eccentricities of the various authors. For example, there isn't a logical reason for testing whether or not SAV supports the device on which the system image file is to be written before checking the syntactic validity of the command. At one point in time, it saved a few words of code.

1. If the SAV command was not issued from a privileged terminal, SAV issues the PRIVILEGED COMMAND error message and exits.

2. The name and logical unit number of the system disk (frequently referred to as the load or boot device) is retrieved from $SYSIZ+2 and $SYSIZ+6. They were left there by BOO, which had to know that information to find the system image file to boot.

   The device name is checked against the names of all the I/O devices in the system. If no match is found, the NOT VALID SAVE DEVICE error message is issued and SAV exits. If this occurs, the locations around $SYSIZ have probably been corrupted. INITL would not have allowed the system to come up if the boot device was not in the system I/O data structures.

3. The SAV command line is obtained. If there is an error, SAV outputs the COMMAND I/O ERROR message and exits.

4. SAV sets bit FE.MXT in the first system feature mask word ($FMASK) and unstops tasks that are not stopped for an event flag or buffered I/O. This should cause all Command Line Interpreters (CLIs) to exit because they should check for FE.MXT being asserted before they stop themselves. As a side effect, other tasks, such as HRC..., will also exit. This makes such tasks inactive when the system is copied into the system image file. Thus VMR can remove and reinstall new copies of those tasks.

   Note that SAV will hang if any CLI does not exit. This is only the first example of a general principle. There are systems that cannot be saved. There is an implicit pact between the system as a whole and SAV to provide a "reasonable" environment. There are some cases SAV cannot win, so why try unreasonably hard?

   This philosophy (justified in part because SAV does not have enough address space to be really paranoiac) is rarely challenged simply because SAV is almost solely used immediately after booting a virgin system image file.

5. SAV makes sure that the pool monitor task (PMT) is inactive. If it is installed, an attempt is made to force it to exit.

6. The command line is parsed. If it is syntactically incorrect, a SYNTAX ERROR message is output, and SAV exits.

7. SAV makes sure that the system image file is big enough to hold everything in main memory that is of value (defined to be secondary pool and everything that has a PCB except device commons). If that is not the case, a COMMON, DRIVER OR TASK ABOVE SYSTEM IMAGE FILE LIMIT error message is printed, and SAV exits.

   Either make the system image file large enough or use the MCR PAR command to determine what is above the system image file size and get rid of it or move it lower.

8. SAV attempts to force installed, resident, read/write commons into their file images. It is assumed that a common will remain checkpointed once it is checkpointed, because there is supposed to be minimal system activity when a system is saved. If this is not true, SAV will hang.

It is not obvious why this step is not done before checking to see if all of main memory will fit into the system image file. If all nonreferenced commons were forced out of main memory, the previous check would succeed more frequently.

9. If any checkpoint files are in use, a CHECKPOINT FILE STILL IN USE ON ddn: error message is output for the first such file, and SAV exits.

   ACS should be used to disable checkpointing on that device.

10. If error logging is still active, an ERROR LOGGING STILL ACTIVE error message is output, and SAV exits.

    Run ELI to stop error logging.

11. If there are any checkpointable commons that are not installed from an LB:, a COMMON name NOT INSTALLED FROM AN LB: error message is output for the first such common, and SAV exits.

    When a saved system image is booted, the task image files of all installed commons and tasks are checked to see if they are still usable (for example, that they have not been deleted or that the disk they reside on is still accessible.) Tasks and commons that have unusable task image files are removed from the system. An obvious way for a task image to be unaccessible then is not to be on the booted (system) device, which is the only accessible device at that time.

    To make it easy to avoid this situation, a convention has been established. If a common or task is installed from any LBn:, it is assumed that its common or task image file will be accessible when the system is booted. Any LBn: was chosen as the criteria, because LB0: is the default device for VMR INS and because LB0: tends to always point at the booted (system library) device. Because LBn: will pass this test, users can run multidisk systems if they are willing to assume the responsibility that all such disks will be accessible when the system is booted.

    The MCR command CBD lists checkpointable commons.

12. If the processor on which SAV is executing is not the only active processor, a PROCESSOR x IS NOT STOPPED error message is output for the first such processor, and SAV exits. Note that the CPU identifier x in the error message is a letter (A, B, C, or D).

    CON DISPLAY FULL FOR CP shows the status of all CPUs. CON OFFLINE CPx or CON OFFLINE ALL can be used to take CPUs offline.

13. If any mountable devices in addition to the system device are mounted, a VOLUME STILL MOUNTED ON ddn: error message is output for the first such device, and SAV exits.

    The MCR DEV command can be used to determine which devices have mounted volumes.

14. If any files are still open on the system device, an OPEN FILES ON ddn: error message is output, and SAV exits.

    This is most likely caused by having a print or batch queue still active. Stop the various despoolers and the queue manager.

15. If the system device is mounted, SAV attempts to automatically dismount it. If the dismount succeeds, a couple of different messages can be output depending on whether or not TKTN is installed. If the dismount fails, an ERROR ATTEMPTING TO DISMOUNT ddn: error message is output, and SAV exits.

16. If any task has I/O active, a TASK name HAS OUTSTANDING I/O error message is output for the first such task, and SAV exits.

    The MCR ATL command can be used to identify such tasks.

    The test is made, because such tasks would hang when the system is later booted and, needless to say, the device is no longer active to cause an interrupt.

17. If any task is active and checkpointed, a TASK name IS ACTIVE AND CHECKPOINTED error message is output for the first such task, and SAV exits.

    The MCR ATL command can be used to identify such tasks.

    The test is made because tasks can be simultaneously installed in more than one system image file. Therefore when the other system image runs, it could checkpoint the task into its task image file. When this system image file is then booted, the task from the other system would be simply loaded into main memory and used (or executed) by the system. This would probably cause a very mysterious crash.

18. If any task is found to be connected to an interrupt vector (the $CINT directive), a TASK name IS CONNECTED TO AN INTERRUPT VECTOR error message is output for the first such task, and SAV exits.

    Abort the task or use some other method to cause the task to relinquish the interrupt vector.

An RSX-11M system cannot tolerate tasks being connected to interrupt vectors because the TCB word T.CPCD is used by both SAV and connect-to-interrupt. This is not a problem in M-PLUS, but the problems of tracking down the interrupt vectors (they are not in the system I/O data structures) and bus affinity are severe enough to postpone a general treatment in SAV until a future release (if ever).

19. If any tasks are installed from some device other than an LB:, a TASK name IS NOT INSTALLED FROM AN LB: error message is output for the first such task, and SAV exits.

    The MCR TAS command can be used to determine from which device a task is installed.

    The rationale behind this check is the same as for having checkpointable commons installed from an LB: and suffers from the same problems.

    Note that VMR INS will install tasks only if their task image files are on LB0:.

20. SAV then disables checkpointing in the system.

21. A test is made to insure that accounting is turned off. If it is active, the message ACCOUNTING IS ACTIVE is sent, and SAV terminates.

22. If secondary pool is not completely within the range of memory to be saved in the system image file, SAV will output the message SECONDARY POOL DOES NOT FIT INTO THE SYSTEM IMAGE and terminate. This check is made because many things are now stored in secondary pool (for example UCB extensions.)

23. If any controllers except those for the system device, TI:, MK:, and II: are online, a PROPER CONTROLLERS AND/OR UNITS ARE NOT OFFLINE error message is output, and SAV exits.

    If any units except those for the system device, TI:, MK:, pseudo devices, and RD: are online, the error message is output, and SAV exits.

    In either case, CON DISPLAY FULL will show which controllers and/or units are online.

    CON OFFLINE ALL should leave only the permissible controllers and/or units online. However, because CON OFFLINE ALL suppresses error messages, the system may be having problems during the offline process and you may be unaware of them. In this case, CON should be used in an attempt to put individual controllers and/or units offline. The problems pointed out by the resulting error messages should be corrected.

TI: and the load device are exceptions, because SAV wants to send QIOs to them. Pseudo devices are exceptions, because they are always redirected to real devices, which can be handled normally. MK: and II: are exceptions, because they are integral parts of a system configuration. RD: is an exception, because it must be online to bring anything else online.

24. At this point all normal operations of the system should be quiesced. However, each main partition's wait queue is checked to insure that no task is still checkpointed into its task image. Note that the checkpoint files have already been checked to make sure that they are empty. SAV also looks at the loader task's receive queue and will output a message if anything is queued to the loader.

    All of the necessary checks on the system have now been made. The checks that follow assume that the system is in a quiescent state.

25. If SAV cannot find the name of the required special driver in the table in PSECT DRVTAB, a NOT VALID SAVE DEVICE error message is output, and SAV exits.

    Except for internal errors (the size of one entry in DRVTAB is wrong or a module is missing) and a corrupted $SYSIZ+6, this check should always succeed.

26. If the desired special driver is found, the UMR usage flag is set and the BAE offset is calculated (for RH and DL devices) and the ring and packet pointers are initialized in case the device is connected to a UDA50 controller. The special driver is then initialized to read in the system image file and is then copied into the system image bootstrap if necessary.

27. The structure level of the volume on which the system is being saved is checked. This prevents "old" SAVs from being used in situations that they might not be able to handle. For example, when a disk supports 65K files per volume, it will have a multiheader index file. SAVs that have not been modified to handle this circumstance should not be used on such disks.

28. The label block of what SAV believes to be the system image file (LBN in $SYSIZ+3 and $SYSIZ+4) undergoes several tests. If it cannot be read in, a LABEL BLOCK I/O ERROR message is output, and SAV exits. If it does not pass a few cursory checks on its integrity as a label block (for example, the file has no header), a $SYSIZ DOES NOT POINT AT SYSTEM IMAGE FILE error message is output, and SAV exits. If all the checks succeed, the number of blocks composing the system image file bootstrap minus one is written into the label block as the "transfer address", and the label block is written back to the file.

Altering the transfer address of the file gives VMR and BOO a simple test for determining if a file is a saved or virgin system image as well as for telling them how long the system image bootstrap is.

29. Context switching is disabled so that tasks the Executive starts before SAV restores the T.LBN's won't get into the active task list and then block all tasks except for SAV. The low order bit of T.STAT is set in every TCB except SAV's. Because this is one of the blocking mask bits, all tasks except SAV are prevented from executing.

30. All copies of a multiuser task have a pointer to the read-only part of the "parent" task image. In a running system, P.LBN is the high byte of the LBN of the first byte of the read-only code. P.LBN+2 is the low order word of P.LBN. P.LBN+4 is zero. In a saved system image file, P.LBN is the high byte of a number which when added to the LBN equivalent of T.LBN will yield the LBN of the read-only segment of the "parent" task image file. P.LBN+2 is the low order word of P.LBN. P.LBN+4 is nonzero to indicate that P.LBN contains a relative number.

31. For every installed task, T.LBN, T.LBN+1, and T.ASTL (if a task is not fixed and inactive) or P.SIZE+2 (if the task is fixed or active) is overwritten with the task's File ID. INS copied the File ID from the task image file label block into the task's header.

    If SAV cannot read a task header during this process, a TASK HEADER READ ERROR message is output, and SAV halts the processor. At this point, the system has been corrupted. Not only have all tasks except SAV been stopped, but some of the P.LBNs and T.LBNs have been converted. Only the part of SAV that brings a system up can reconvert those pointers to the form used in a running system. You must delete the system image file and use VMR to apply the appropriate SYSVMR.CMD file to a copy of the corresponding virgin system image file.

32. If SAV was invoked with the /WB switch (write hardware bootstrap), the special driver is initialized to read the system image bootstrap into low main memory and is copied into the hardware bootstrap block (see $BTBLK in the module SAVBOT). The LBN of the system image bootstrap is written into locations 4 and 6 of the hardware bootstrap block and the hardware bootstrap block is written into LBN 0 of the disk.

33. The special driver in SAV is then set up to write all of main memory into the system image file. This includes setting the ring and packet pointers in SAVCM in case the SAV device is the UDA50.

34. SAV calls the Executive (and consequently the appropriate drivers) to put the controllers and units of the system device, TI:, MK:, and II: offline. For several reasons, it is assumed that there will be no delay in this process and that it will succeed (see WIPVC in the module SAVST for the details). The controllers and units for programmable memory boxes are simply marked offline.

35. CO: is redirected to TI: (see below for the special action that is taken if this is not still true when the system comes up) and all terminals are set nonprivileged and logged out.

36. The software volume valid bit is cleared for the system device. After this, SAV cannot issue any additional QIOs because they will be rejected by the driver.

37. If the clock has a CSR address, it is stopped. The processor's mapping registers are saved on SAV's stack.

38. Bytes 0 through LOWSIZ*2 are copied into a buffer in SAV. The system image bootstrap (see $VBN1 in module SAVVB1), which includes the special driver primed to read in the system image file, is copied into low physical memory.

39. The LBN of the system image file (the system image bootstrap) is copied into physical locations 4 and 6 and into physical locations BTADD+4 and BTADD+6.

40. The LBN and offset within the block of SAV's buffer for low main memory are written into physical locations 10 and 12. This allows VMR to find low main memory in the system image file.

41. SAV calls its copy of the special driver to copy all of main memory into the system image file. The LBN of the system image file is found in locations 4 and 6.

42. When the driver is finished copying main memory, it transfers control to location 0, where the unused system image bootstrap resides. Note that at this point the CPU is left in 22-bit mode with D-space, and the unibus map turned off.

IX. What SAV does in Detail (Restoring the System)

It is at this point that control passes to the saved system from BOO, SAV, or from a hardware bootstrap. Note that this point can be entered with the CPU in either 16- or 22-bit mode. If the system has just been booted by the hardware bootstrap, the CPU will be in 16-bit mode. If control got here via the SAV or BOO commands, the CPU will be in 22-bit mode.

1. If the system has been booted by the hardware bootstrap, the boot block (LBN 0) of the disk has just been read into main memory locations 0 through 777. The purpose of this boot block is to read in the system image bootstrap. The boot block was written to the disk by the /WB command.

   The boot block relocates itself to a place in memory that is well above the size of the system image bootstrap. This is currently calculated by adding 1000 to the size of the system image bootstrap. Once the boot block has been relocated, a copy of the special driver is used to read in the system image bootstrap. For MASSBUS, RK611/711, and UDA50 devices, a simplified special driver is used because of size constraints in the boot block (for example, the boot block can only be 256 words long.) These simplified drivers take advantage of the state of the disk hardware left setup by the hardware bootstrap.

2. The system image bootstrap picks up the LBN of the file to read into main memory from locations BTADD+4 and BTADD+6 and deposits them into locations 4 and 6 for use by its special driver in reading in the rest of the system image file.

   If the system image bootstrap had been read into main memory by BOO, the contents of BTADD+4 and BTADD+6 would have been supplied by the module BOTPH2.

   If the system image bootstrap had been read into main memory by the disk's boot block (see $BTBLK in the module SAVBOT), the contents of BTADD+4 and BTADD+6 would have been supplied by the disk's boot block. The LBN of the hardware bootable system is stored in relative locations 4 and 6 of the disk's boot block. When control is transferred to the hardware boot block from the hardware bootstrap, the hardware boot block relocates itself to BTADD and then reads the system image bootstrap into locations 0 to BTADD.

3. The routine $VBN1 is then entered and a number of things happen to get the system back on the air. Figure 3 gives a graphic description of the steps executed in $VBN1. The $VBN1 process involves the following steps:

1. The processor's mapping registers are set up. Kernal APR6 is specifically set up for use by certain special drivers and User APR5 and APR6 are set up to map to SAV itself.

2. Register SR3 is set for 22-bit and UMRs. If a nonexistent memory trap occurs, the processor does not support 22-bit addressing. For M-PLUS V2.0, this is an invalid condition.

3. A special test is made for a DL (RL01/02) device connected to an RLV12 controller. If this test is successful, SAV is running on a PDP-11/23-Plus processor which has 22-bit addressing and a BAE register.

4. $VBN1 now looks at the name of the special driver contained in the driver block further up in the $VBN1 code. If the driver is the common driver (SAVCM) and the boot device controller is an RK611 or a UDA50, then SAV assumes that UMRs are required to read in the image. If the special driver is not SAVCM, then UMRs are used by default.

5. SAV now knows that the boot device controller is either an RH11 (no RHBAE) or an RH70. If the SYSID register does not exist, SAV must be executing on a PDP-11/24 or PDP-11/44 processor, which means that the controller is an RH11 and UMRs are required. If the SYSID register exists, SAV is executing on a PDP-11/70. However, the controller may not be an RH70. SAV then determines which controller is used and calculates the RHBAE offset by testing for the last controller register that responds and subtracting the CSR address from it if the controller is an RH70.

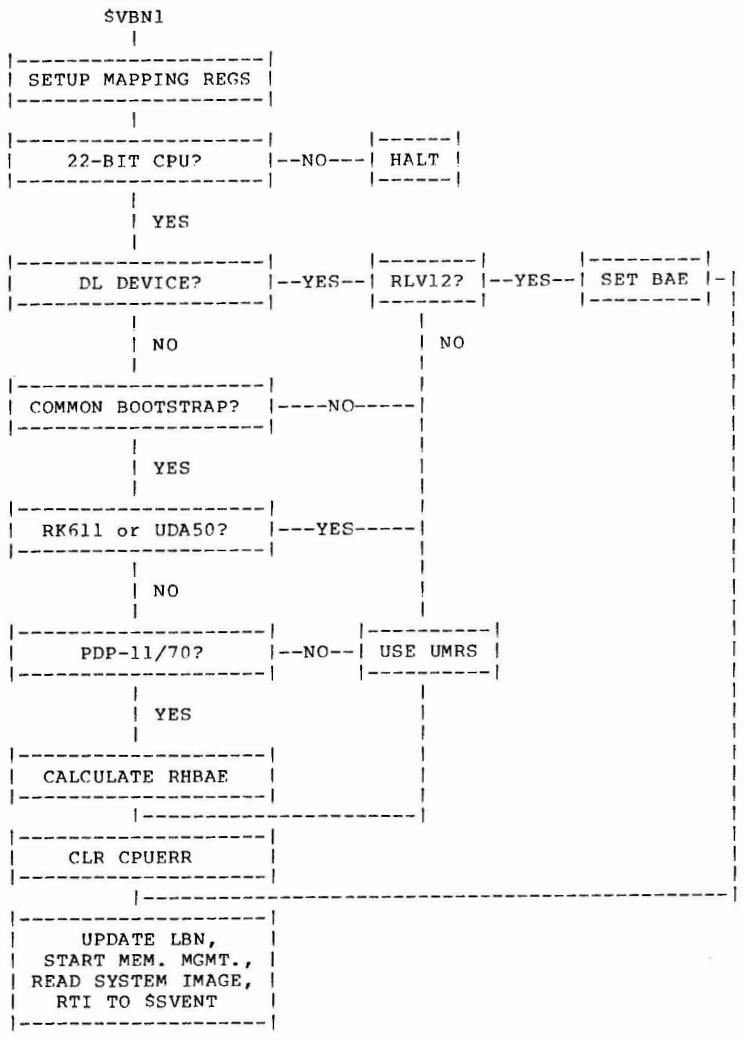
6. If this is a 22-bit CPU and the boot device controller is not an RH70, set register R2 nonzero to indicate that UMRs are required for the special drivers.

7. The CPUERR register is cleared to remove any residual error indicators. Next the LBN of the system image is updated to point to the next area of the image to be read in (currently VBN3).

8. The memory management unit is then enabled (SR0).

9. Control then falls into the system image bootstrap's special driver, which reads into main memory (starting just above the system image bootstrap) the remainder of the system image file.

FIGURE 3 -- FLOW DIAGRAM FOR $VBN1



4. The system image bootstrap then passes control to SAV at location $SVENT in the module SAVE. There the LBN of the system (found in locations 4 and 6), the physical unit number, the physical device name, and CSR address of the load device (passed to SAV in R4, R1 and R5) are saved. Bytes 0 up to LOWSIZ*2 are restored from SAV's buffer, and the mapping registers are restored.

   SAV currently initializes user data space by restoring it from its own user I-space. This implies that SAV cannot be built with user I/D space support unless the code is changed to save and restore both sets of registers.

   When the UMRs are restored, three distinct algorithms may be used. First, if this CPU has UMRs and the saved system had UMRs, then load the UMRs with the values saved and set HF.UBM in the hardware feature mask word ($HFMSK). Second, if this CPU has UMRs and the saved system didn't, then load only the first five UMRs and set HF.UBM. Third, if this CPU has no UMRs (PDP-11/23-Plus), then simply clear HF.UBM.

5. To increase the system's transportability, test the FPP hardware and see if it is present. If it is, clear HF.FPP, otherwise, set the bit. Then test to see if the booted processor has the CIS hardware. Note that this check uses a special stack because the CIS hardware requires a minimum of 64 words of stack space. Set HF.CIS if the CIS hardware is present.

6. If the system is a multiprocessor system, the Interprocessor Interrupt and Sanity Timer (IIST) is interrogated to determine the number of the booted processor. If the IIST exists (if it does not, the booted processor is considered to be CPU A), the processor is mapped to the appropriate processor-dependent, low main memory context. The IIST is marked online, and its interrupt vector is established.

   Note that a dual processor system consisting of CPUs A and B cannot be booted and then run as a dual on CPUs C and D. Each processor requires its own processor-dependent context. The processor finds its context by determining its absolute processor number by interrogating the IIST and then looking in the appropriate part of the CPU partition. A dual consisting of CPUs A and B does not have processor-dependent context areas for CPUs C and D. Therefore the only truly transportable multiprocessor system is either a single- or a quad-processor system.

7. If the booted processor is not the processor on which

the system was saved, SAV's context is logically transferred to the booted processor. This involves actions such as logically making SAV the current task on the booted processor, marking the other processor as stopped, exchanging trap vectors, and the like.

8. If the processor on which the system was saved had a free running clock (a line frequency clock with no CSR), SAV checks for a programmable clock or a line frequency clock with a CSR. If one is found, it is used as the system clock. Unless this is done before a system was saved on a processor with a free running clock, it could not be run on a system without a free running clock, because no attempt would be made to find (enable the interrupts for) another type of clock. If no clock with a CSR is found, it is assumed that the booted system also has a free running clock.

   Note that SAV will hang when it waits for the first redirect of the system device to occur unless the system has a ticking clock.

   If the booted processor has the same type of clock as the processor on which the system was saved, that clock becomes the system clock. If that is not so, it must be one of two cases:

   - The processor on which the system was saved had a line frequency clock, and the booted processor has only a programmable clock. SAV runs the programmable clock at line frequency. Note that if the programmable clock has not been properly installed (had a line frequency signal run to its Small Peripheral Controller slot), it will not tick if run at line frequency.

   - Or, the processor on which the system was saved had a programmable clock, and the booted processor has only a line frequency clock. SAV treats line frequency as 50 or 60 hertz, depending upon the SYSGEN response to the system hertz question.

   Note that all items in the clock queue are scheduled by ticks. Not finding the type of system clock with which the system was saved may cause SAV to change the number of ticks per second for the booted system from that of the saved system image file. SAV makes no attempt to alter the clock queue to reflect any change in the number of ticks per second. All the events already in the clock queue will occur in the correct relative relationship, but the absolute times when they occur will be off by an undetermined factor. However, entries added to the clock queue after the system is booted will occur at the correct absolute times.

9. By examining the contents of the Floating Point Processor (FPP) trap vector and testing for the existence of the PIRQ register, the FPP trap vector is altered to correctly handle the processor's FPP.

10. To increase the system's transportability, the Executive contains a table of CSR addresses. These addresses are typically those of parity memory modules.

    SAV checks to see if those addresses exist in the I/O page of the booted processor. If not, the table is altered to point to a main memory location. This allows the Executive to contain unconditional code that manipulates those CSRs.

11. A series of tests is then made to determine the type of CPU that SAV is executing on. At the present only the PDP-11/74, PDP-11/70, PDP-11/44, PDP-11/24, and PDP-11/23-Plus are fully supported. The CPU model number is stored in the Executive as a decimal value.

12. Main memory is then sized. This consists of writing a zero into every memory location above the area written into by the boot process (so the parity memory bits are set to a known state) and noting where memory first does not exist.

    SAV always sizes memory because after the CPU mapping has been set up, SAV might be able to find memory that the special driver could not when it was DMA'ing in the system image. For example, we might find UNIBUS memory on the PDP-11/44.

    The last (highest) main partition in the system that is not a device common is automatically expanded to cover all memory that is found, and $SYSIZ is updated. The partition should not be expanded if it is a CPU or secondary pool partition.

    If the booted processor has MKAll main memory, SAV will find only the memory boxes that have their "force panel" switches set to force the boxes' starting addresses to be determined by the control panel's starting address switches. Fither the M9312 bootstrap or BOO will have forced the programmable boxes out of the main memory address space.

13. SAV makes sure secondary pool exists and makes sure that it is within the available memory. Note that this routine must precede the one that brings the console and system disk online, since data structures for those devices are created in secondary pool.

14. The system I/O data structures are then searched for a device with the same name, same CSR, and same physical

unit number as the booted device. If such a data structure is not found (for example, the system was booted on DB2:, but data structures exist only for DB0: and DB1:) or, if the booted device is not on the UNIBUS run of the booted processor, SAV attempts to output (the console may not exist or function) a BOOTED DEVICE NOT IN SYSTEM - dd nnn mmmmmm x error message, where dd, nnn, and mmmmmm are the device name, physical unit number and CSR address for which SAV was looking, and x is the name of the CPU on which SAV thinks it is running. The CPU is then halted with a number greater than 120000 in R0.

15. If the load device exists in the system data structures, SAV, if necessary, alters the data structures to show that the port for the device is switched to the booted CPU and checks for the driver being loaded. If the driver is not loaded, SAV attempts to output a SYSTEM DISK DRIVER NOT LOADED error message and halts the CPU with a number greater than 120000 in R0.

    Boot another system and use VMR to load the driver for the system disk.

16. The Unit Control Block (UCB) for the booted device is redirected to itself. Its software volume valid bit is set, and it is marked as public if the system is a multiuser protection system.

    An attempt is made to bring the booted device online, which, if the operation is successful, accomplishes several things. The interrupt vector(s) will be set to point to the driver interrupt entry point(s), common interrupt routine, or Interrupt Control Block(s) as appropriate. If the device can operate on a MASSBUS, it is determined if an RH11 or RH70 is involved.

    To do this, M-PLUS uses a different technique than does RSX-11M. In M-PLUS, the driver has a special entry point for going online, and the data structures show where the RHBAE register should be if it exists. (If the RHBAE register exists, it is an RH70.) In RSX-11M, SAV contains a table of device names that are associated with MASSBUSes and their standard number of CSR registers.

    If the device cannot be brought online, SAV outputs a BOOTED DEVICE CANNOT BE BROUGHT ONLINE error message, and the system is halted with a number greater than 120000 in R0.

    Short of a bug in the system disk driver or a badly corrupted system, this error message should never be output.

17. If a directive partition exists and is not loaded (the APR mapping for the partition in $DRAPR is zero), SAV attempts to output a DIRECTIVE PARTITION UNFIXED OR NONEXISTENT error message, and the processor is halted with a number greater than 120000 in R0.

    This check is accomplished in two ways. If the contents of $SYSIZ are greater than the contents of $DRAPR, the directive common may have been completely read into main memory. Or, the PCB that is associated with the directive common can be found, and the directive common's upper bound can be checked against the contents of $SYSIZ.

18. The system disk is sized by a module stolen from reconfiguration. Sizing allows the system to adjust to the booted device being slightly different from the saved device. For example, BRU can be used to copy an RK06 disk that contains a hardware-bootable saved system image file to an RK07. The RK07 will be hardware bootable after the copying, because the sizing of the system disk will adjust the UCB for the system device to that of an RK07.

    This implies that SYSGEN questions about what type of disk drives are on what controllers are useless except for virgin systems (because virgin systems do not size disks). Actually, they are even useless for virgin systems except for the RK06/07 case. The RK06/07 driver must know which type of unit is being used, because an RK06/07-type bit in the CSR must be properly asserted or deasserted to read or write the unit.

19. Pool is now sized if the system supports the pool monitoring task (PMT).

20. SAV then determines which unit should be CO:, the console. If CPUn is booted, YLn (the nth DL11 terminal interface) is the default console. In a single processor system, this is TT0:.

    When the system is saved, CO: is redirected to TT0: by SAV. If CO: is not redirected to TT0: when the system is booted (that is, someone uses VMR to alter the redirection), SAV attempts to use the unit to which CO: is redirected as the console. This allows a system to be booted on a configuration where TT0: does not work.

    If the device to which CO: is redirected does not make a good console device (for example, it is not a terminal device, or it does not exist in the booted configuration), SAV attempts to use the default terminal.

If no device makes sense as a console terminal, SAV halts the processor with a number less than 120000 in R0 (currently 1).

21. When the console terminal is determined, SAV alters the system data structures to show the console terminal as logged in, nonslaved, and privileged. CO: and CL: are redirected to it, and it becomes TI: for SAV.

22. Because the module SAVINS can potentially queue commands to MCR, a check is made to make sure that the MCR dispatcher is installed. If it is not, an MCR IS NOT INSTALLED error message is output, and the processor is halted.

23. The system ID message is output.

24. The redirection of the pseudo devices SY:, LB:, and SP: to the booted device is accomplished by direct manipulation of the data structures. This must be done so the MCR dispatching action for the RED and MOU commands that follow will operate from the booted disk.

25. Tasks are then "reinstalled." This basically consists of undoing SAV's alterations to the various P.LBNs and T.LBNs and, if necessary, "rebinding" the tasks' Logical Unit Table (LUT) entries (Logical Unit Numbers, LUNs) to the UCB addresses of this system and the tasks' headers to the PCB addresses of this system. (For information regarding the logical structure of the booted disk and task images, see the appendices of the IAS/RSX-11 I/O Operations Reference Manual and the RSX-11M/M-PLUS Task Builder Manual.)

    SAV first finds and validates the home block of the booted device. After validation, the home block is used to find the index file header, which is also validated.

    For each installed task, the task's File ID is retrieved from the appropriate TCB. Given that and the index file header, the file header of each task can be found. The file header is validated and, if any harm has come to it (for example, the task image file was deleted), an error message is output, and the task is removed.

    Given an intact task image file, the label block is read and the LBN of the task image is determined and is rewritten into T.LBN and T.LBN+1 of the TCB.

    If the task has a read-only segment, the P.LBN that points to the read-only segment is converted from a number relative to the T.LBN of the task image file that contains the read-only segment to the LBN of the

segment. P.LBN+4 is used as a flag to determine if the conversion has already been performed.

If the first word of each of a task's LUT entries is not a valid UCB address of the booted system, Assign LUN Directives are done using the static LUN assignments of the task's label block. The resulting UCB addresses for the booted system are written into the first words of the LUT entries.

Note that while any dynamic LUN assignments made by the task are lost, the UCB addresses that SAV derives are the same UCB addresses that INS would use if the task were to be manually reinstalled using the same global logical assignments as occur in the system image file (that is, if STARTUP.CMD does not establish any global logical assignments). Some sort of reinstallation of the task must be done or the system will crash when a LUT's "UCB address" is used as a UCB address by the system. A task can have the "wrong" UCB addresses in its LUT if the task is simultaneously installed in more than one system image file.

If the label block device assignment for a LUN is not for a device that is in the booted system, a (WARNING) NONEXISTENT LUN ASSIGNMENT FOR TASK name error message is output and no assignment is made for the LUN (that is, a zero is written into the first word of the LUT entry). The task will get an error if it tries to use the LUN without assigning a device to the LUN.

There is a similar problem with the mapping windows in the task header. If the task is simultaneously installed in more than one system image file, the PCB addresses in the windows may not be for the booted system. If SAV discovers such a case, it rewrites the addresses to the appropriate PCB addresses of the booted system. The correct PCB addresses are derived from other system data structures (T.PCB and T.PCBV) so even the effect of installing a task with a /PAR=partition-name is preserved.

There is one case that cannot be covered. If the task is active and has dynamically mapped regions, a TASK ACTIVE IN ANOTHER SYSTEM, TASK REMOVED - name error message is output, and the task is removed.

If the partition in which the task should execute has shrunk so much (because of the small amount of main memory on this processor) that the task can no longer fit into the partition even if nothing else is in the partition, a TASK TOO BIG FOR PARTITION, TASK REMOVED - name error message is output, and the task is removed.

If the task image for SAV has been damaged (deleted,

for example), the error message SYSTEM MAY NOT BOOT CORRECTLY is output in addition to the message about removing SAV. If the system does not have enough main memory to allow both SAV and MOU to reside simultaneously in main memory, the system will now hang. This is because the system needs to checkpoint SAV to get MOU into memory to mount the system disk, but SAV has been blocked from checkpointing, so it cannot overwrite the part of the disk where SAV.TSK used to be.

SAV should handle corrupted task images better. There is no need for SAV to worry about deleted task image files if the task is fixed in main memory (for example, the loader).

For each common in the common block directory list that has a task image file, convert its File ID (left in P.FID1, P.FID2, and P.FID3 by INS) into a LBN. This allows BRU and DSC to compress disks containing commons as well as task images.

SAV makes sure that each common task image file is not deleted. If one has been, the system is prevented from trying to checkpoint the common into the task image file and a SYSTEM MAY NOT WORK - CORRUPTED FILE FOR COMMON name error message is output for each affected common.

26. After all the reconverting, the tasks are unblocked, and SAV may no longer be the only running task in the system.

27. The system clock is started after all tasks are reinstalled to prevent the possibility of the Executive finding a clock queue entry for a task to be run on the first clock tick (this could result from a RUN command in VMR). If that occurred, the Executive clears the task's TS.EXE bit, which would make $INSTK look in P.WAIT rather than T.EFLG for the task's File ID. The result would be a bad T.LBN for the task.

    After saving the time the system was booted for RMDEMO, the system clock is started.

28. After reenabling system checkpointing, a check is made to see if the booted processor had enough physical memory to hold all of the system image file's important structures. Important structures are defined to be those that have PCBs and are not device commons. If the check fails, a SYSTEM MAY NOT WORK - LARGER THAN MAIN MEMORY error message is output. Note that secondary pool is checked prior to this.

    The system will work if those structures that were not

read in from the system image file can be logically removed before the system makes an attempt to reference them (for example, unload any loaded drivers that might not have made it into main memory.)

29. Just to give the user a record of what is going on, commands for redirecting the pseudo devices SY:, LB:, and SP: to the system device are queued to MCR. This has the side effect of printing them on the console.

    As an historical note, there is another device name that is treated specially. WK: is meant to be the work file device for such tasks as MAC and TKB. The intent is to identify a LUN that will have heavy I/O usage. Assigning the LUN to a fast device (for example, a fixed head disk) can improve the performance of a system.

    The idea has been partially implemented. VMR automatically creates a global logical assignment for WK: to LB: for all system images. Unfortunately, WK: is not generally referenced in task-build command files. Therefore if WK: is to be used, the task must be installed and the work file LUN must be manually identified and reassigned to WK:. There is little point in doing this, because the work file LUN could just as easily be assigned directly to the real device rather than to WK:.

    There are two reasons for WK: being a logical device name rather than a pseudo device. Pseudo devices must always be redirected to physical devices (with the exception of TI:, which is special cased in the Executive). When the system is booted, there is only one disk device available, the system device. Therefore the pseudo devices are redirected to it. Because it is illegal to redirect devices that are redirected to mounted devices and because the system device is automatically mounted, if WK: were a pseudo device, it could not later be redirected to the real work file device.

    If the tasks that use WK: do runtime LUN assignments, the logical name WK: provides more flexibility than a pseudo device. Each user can independently establish his own work file device by simply making WK: a local logical name. This can prevent running out of space on the one system work file device due to the aggregate demand on that one device of all running tasks that use work files.

30. The system disk is mounted. The defaults that MOU uses can be overridden if the /MOU switch was used when the system was saved. In must cases, overridding the defaults can improve system performance. For instance,

see the discussion of the /LRU and /WIN switches in the discussion of the MOU command in the RSX-11M/M-PLUS MCR Operations Manual.

31. The startup indirect command file is initiated. If the default file specification is used, SAV inserts the hardware name of the boot device to avoid the problem of a global logical assignment for SY: that is not directed to the booted device.

32. Finally, if the system supports the pool monitor task (PMT), PMT is initiated if it is installed.

RSX-11M-Plus V2.0 Boot Problem

Steve Gale

The Computer Communications Group
Bell Canada
160 Elgin Street
Ottawa, Canada K1G 3J4

1.0 THE PROBLEM

It has been discovered that RSX-11M-PLUS V2.0 will not boot correctly on our computer systems which use CDC disks and S.I. 9400 disk controllers. The problem has been isolated to a minor difference between the S.I. disk controller and the equivalent DEC RH (RP04) disk controller. The S.I. controller does not emulate the controller registers exactly as defined in the DEC RP04 disk controller specifications. In particular the LOOK AHEAD register (RPLA) is implemented as read/write register by S.I. whereas DEC implements this register as a read only register.

The RSX-11M-PLUS V2.0 operating system utilizes a common disk driver to boot all large disk systems. Since the DEC large

disk controllers (RK/RH type controllers) use the same register addresses the bootstrap disk drivers must dynamically determine the type of disk from which the system is being booted. In order to differentiate between the RK and RH disk controllers the disk driver attempts to write a one (1) to the DESIRED CYLINDER register (RKDC) in the RK disk controller. Subsequently the content of this register is checked to see if it is indeed a one (1). If so the RK type of controller is assumed present. On systems which have a DEC RH type controller the RKDC register corresponds to the RPLA register which is a read only register. Therefore the attempt to write to this register fails and an RH type controller is assumed present. However since the S.I. controller has a read/write RPLA register the write succeeds. Therefore the driver attempts to continue the boot assuming an RK type controller which results in a system crash.

2.0 A SOLUTION

In order to correct the boot problem the simplest fix is to disable the check for a RK disk controller so that the bootstrap disk drivers always detect the RH type of controller is present. This is done by changing the write one (1) to the RKDC register to a write zero (0). Therefore an RH type controller is always detected since the register does not contain the desired value (1). After doing this the system will not be bootable from RK06/RK07 disks. Bootstrapping from other devices supported by V2.0 will not be affected. The bootstrap programs for V2.0 utilize a two stage boot and both stages must be modified. In order to make the required modifications the distribution disk must be altered both at the hardware and software bootstrap levels.

There are five areas which must be altered in order to disable all RK disk controller checks. First of all the hardware primary bootstrap must be altered. This bootstrap is located in the bootstrap block on the distribution disk (virtual block 1 of [0,0]INDEXF.SYS). Next the secondary hardware bootstrap must altered. This bootstrap is located in virtual blocks 2 thru 4 of the baseline system image file [2,54]RSX11M.SYS. Thirdly the baseline system software boot task ([2,54]BOO.TSK) must be modified in order that the system resulting from a subsequent syssen will be software bootable in order to SAVE it. Finally three modules in the SAVE command object libaray ([1,24]SAV.OLB) must be altered in order that subsequent syssens will produce a SAV.TSK capable of saving a hardware bootable system. Syssens will also create a new BOO.TSK which is capable of software booting V2.0. Note that in order to make the changes to the baseline system it is necessary to have an existing RSX-11M/11M-PLUS system which resides on a disk other than the destination disk for the V2.0 distribution kit. The following points outline the steps to take to make the required changes.

1) Boot the standalone BRU from tape. Note that you should use the BRU distributed with V2.0 as BRU tape formats may not be compatible between V1.0 and V2.0. See the V2.0 release notes for details.

2) Copy the baseline system to the destination disk.

3) Boot an existing system so that you can alter the baseline hardware/software boot disk drivers.

4) ZAP the bootstrap block ([0,0]INDEXF.SYS)on the V2.0 distribution disk to disable disk type checking in the hardware primary boot.

5) ZAP the V2.0 baseline system image (RSX11M.SYS) to disable disk type checking in the hardware secondary boot. The baseline system will be hardware bootable after performing this step.

6) ZAP the V2.0 baseline system boot task (BOO.TSK) to disable disk type checking during subsequent software boots.

7) Boot the baseline system and proceed with the syssen procedure until asks if you want to edit any files before assembling (question AE030). Answer YES to this question.

8) Create the correction files SAVBOT.COR, SAVVB1.COR and SAVCM.COR in UFD [12,60].

9) Apply the corrections to SAVBOT.MAC, SAVVB1.MAC and SAVCM.MAC in UFD [12,10]. Do not destroy the originals.

10) Assemble the corrected SAVBOT.MAC, SAVVVB1.MAC and SAVCM.MAC files. Replace modules SAVBOT.OBJ, SAVVB1.OBJ and SAVCM.OBJ in [1,24]SAV.OLB with the new modules.

11) Continue the syssen until completion.

12) Software boot the new system and SAVE it as usual.

After completing the above procedure system produced will be bootable without further changes. Subsequent syssens will also produce bootable systems.

NOTE

Subsequent applications of AUTOPATCH to the distribution kit may result in a new version of SAV.OLB being placed in UFD [1,24] and/or modified versions of the sources SAVBOT.MAC, SAVVB1.MAC and SAVCM.MAV being placed in UFD [12,10]. If this is the case the corrections must be reapplied to the sources and the object libary updated again to ensure correct booting of the any newly generated system.

Following is a sample dialog of the above procedures.

3.0 SAMPLE DIALOG

```
>;
>;
>;  BOOT STANDALONE BRU
>;
>;
+MM

RSX-11S Mapped Standalone system V01


RSX-11M/RSX-11M-PLUS Standalone Configuration and Disk Sizing
 Program
        .
        .
        .       SOME OTHER JUNK
        .
        .

Enter first device: MM0:

Enter second device: DB1:

Hit RETURN and enter date and time as 'TIM HH:MM MM/DD/YY'

TIM 8:30 08/23/82

RUN BRU
BRU>/INIT MM: DB1:
BRU - STARTING TAPE 1 ON MM0:

BRU - END OF TAPE 1 ON MM0:

BRU - COMPLETED

BRU>^Z
>;
>;  BOOT AN EXISTING SYSTEM
>;
+DB

 RSX11M+ V01  BL6      768K
>RED DB:=SY:
>RED DB:=LB:
>RED DB:=SP:
>MOU DB:RSX11MPBL6/ACP=DB0ACP
>@DB:[1,2]STARTUP
        .
        .
        .
@<EOF>
>;
>;  MOUNT V2.0 DISTRIBUTION DISK
>;
```

```
>MOU DB1:RSX11MPBL10/UNL
>;
>;
>;  ZAP THE BOOTSTRAP BLOCK
>;
>ASN DB1:=SY:
>SET /UIC=[1,1]
>RUN $ZAP
ZAP>[0,0]INDEXF.SYS/AB
_0:134/
000:000134/ 000001
_0
_X
>;
>;  ZAP SECONDARY BOOTSTRAP
>;
>SET /UIC=[2,54]
>RUN $ZAP
ZAP>RSX11M.SYS/AB
_0:2260/
000:002260/ 000001
_0
_0:2714/
000:002714/ 000001
_0
_X
>;
>;  ZAP THE BASELINE BOO.TSK
>;
RUN $ZAP
ZAP>BOO.TSK
_2:132704/
002:132704/ 000001
_0
_X
>;
>;
>;
>;  BOOT THE ZAPPED BASELINE SYSTEM
>;
>;
+DB1

  RSX-11-M-PLUS V2.0  BL10    512.K System:"BASLIN"
>RED DB1:=SY:
>RED DB1:=LB:
>RED DB1:=SP:
>MOU DB1:"RSX11MPBL10"
>@[2,54]BASTART
>;
>;       BASTART DIALOG
>;
@ <EOF>
>SET /UIC=[200,200]
>@SYSGEN
```



```
          .
          .
          .       SYSGEN DIALOG
          .
          .
>;
>;
>* AE030   Do you wish to pause to edit any files before
          assembling [Y/N D:N]: Y
>;
>; SYSGEN wil now pause to let you edit any files before
 assembling
>; the Executive and drivers.
>;
AT.TO -- Pausing. To continue type "UNS AT.<ESC>"
>;
>;
>;  INSTALL TASKS NEEDED TO MAKE CORRECTIONS
>;
>;
>INS $SLP
>INS $EDT
>;
>;  CREATE SOURCE FILE CORRECTIONS
>;
>;
>UFD DB1:[12,60]
>SET /UIC=[12,60]
>EDT SAVBOT.COR
Input file does not exist
[EOB]
*I
   OU:SAVBOT.MAC;2/AU/-BF=IN:[12,10]SAVBOT.MAC;1
   -22,,/;SG001/
   ;       STEVE GALE 04-MAY-82
   ;       SG001 -- DISABLE RK CONTROLLER CHECK
   ;                RK TYPE DISKS WON'T BOOT NOW
   ;
   -163,163,/;SG001/
           MOV     #0,RKDC(R5)     ;;; TEST FOR AN RK/RH
   /
   ^Z
[EOB]
*EXIT
DB1:[12,60]SAVBOT.COR;1  9 lines
>EDT SAVVB1.COR
Input file does not exist
[EOB]
*I
   OU:SAVVB1.MAC;2/AU/-BF=IN:[12,10]SAVVB1.MAC;1
   -29,,/;SG001/
   ;       STEVE GALE 23-AUG-82
   ;       SG001 -- DISABLE RK CONTROLLER CHECK
   ;                RK TYPE DISKS WON'T BOOT NOW
   ;
   -158,158,/;SG001/
           MOV     #0,RKDC(R5)     ;;; SETUP FOR RK/RH TEST
   /
   ^Z
```

```
[EOB]
*EXIT
DB1:[12,60]SAVVB1.COR;1  9 lines
>EDT SAVCM.COR
Input file does not exist
[EOB]
*I
   OU:SAVCM.MAC;2/AU/-BF=IN:[12,10]SAVCM.MAC;1
   -34,,/;SG001/
   ;     STEVE GALE 04-MAY-82
   ;
   ;     SG001 -- DISABLE RK CONTROLLER CHECK
   ;              RK TYPE DISKS WON'T BOOT NOW
   ;
   -264,264,/;SG001/
            MOV     #0,RKDC(R5)      ;;; TEST FOR AN RK/RH
   ;**-10
   /
   ^Z
[EOB]
*EXIT
DB1:[12,60]SAVCM.COR;1  10 lines
>;
>;
>;  APPLY SOURCE FILE CORRECTIONS
>;
>;
>SET /UIC=[12,10]
>SLP @[12,60]SAVBOT.COR
>SLP @[12,60]SAVVB1.COR
>SLP @[12,60]SAVCM.COR
>;
>;
>; ASSEMBLE CORRECTED SOURCES
>;
>;
>SET /UIC=[1,24]
>MAC SAVBOT=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[12,10]SAVBOT
>MAC SAVVB1=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[12,10]SAVVB1
>MAC SAVCM=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[12,10]SAVCM
>;
>;
>; UPDATE OBJECT LIBRARY
>;
>;
>LBR SAV.OLB/RP=SAVBOT,SAVVB1,SAVCM
   MODULE "SAVBOT" REPLACED

   MODULE "SAVVB1" REPLACED

   MODULE "SAVCM"  REPLACED
>;
>;
>;  CLEANUP OBJECT FILES
>;
>;
>PIP SAVBOT.OBJ;*,SAVVB1.OBJ;*,SAVCM.OBJ;*/DE
>;
>;      CONTINUE SYSGEN
>;
```



```
>UNS AT.
        .
        .      REMAINDER OF SYSGEN DIALOG
        .
@ <EOF>
>;
>;
>;  BOOT THE NEW SYSTEM
>;
>;
>BOO [1,54]RSX11M

XDT: 10

XDT>G
 RSX-11M-PLUS V2.0  BL10

>TIM 23-AUG-82
>;
>;
>;  SAVE THE NEW SYSTEM
>;
>;
>SAV /WB


 RSX1-11M-PLUS V2.0  BL10   512.K System:'RATS  '
>RED DB1:=SY:
>RED DB1:=LB:
>RED DB1:=SP:
>MOU DB1:'RSX11MPBL10'
>@[1,2]STARTUP
        .
        .
        .                          HURRAY IT WORKS
        .
        .
        .
@ <EOF>
>
```

Adding User Translation Tables to MTAACP

B.Z. Lederman

I.T.T. World Communications
New York, New York 10004

One of the enhancements to RSX-11M V4.0 is the ability to insert translation tables into the magnetic tape acp (MTAACP). Most of the information required to do this is in the I/O Operations reference manual, appendix G, section G.6.3, but in adding my own routine I found a few points which could use some clarification. Between my own efforts and some advice from TSC, I offer the following suggestions.

First, it seems desireable to specify a PSECT for the user written routines: the values given in the following example work properly. Also, the appendix states that the translation tables should be 256. bytes long (one translation for each possible 8 bit code), but in the example given, the output translation table is shorter. You should always make your tables the full 256 bytes long: otherwise, sooner or later, you are going to get a tape with characters you didn't expect, and MTAACP could violate it's address limits and abort, locking up your tape drive. Similarly, on output, you may think you are dealing only with 7 bit ASCII, but eventually someone will come up with what looks like ASCII text which has the 8th bit set (maybe to get past the terminal driver): as may be seen in my example, I simply entered the first 128 characters in the table, and then used the editor to duplicate this for the second 128 characters.

```
        .TITLE  USER1
        .IDENT  /V1.2/
;
        .NLIST  BEX
;
;       A User written translation routine for MTAACP
;
;       Translate XXX tapes to ASCII
;
;       B. Z. Lederman I.T.T. World Communications 12-August-82
;
        .PSECT  USER1, RW, I, GBL, REL, OVR
;
;       Entry point: this must be a label MTA understands.
;
USER1::
        MOV     #TBLPTR,R0
        RETURN
```



```
;
TBLPTR: .WORD   BDTASC
        .WORD   ASCBDT
;
;       Translation tables: XXX to ASCII.
;
BDTASC:
        .BYTE   ' ,' ,'5,' ,'<,' ,'9,' ,'*,' ,'#,' ,',,' ,'.
        .BYTE   ' ,'>,' ,'),' ,'4,' ,'&,' ,'8,' ,'0,' ,':,' ,'=
        .BYTE   ' ,'3,' ,'+,' ,'@,' ,'?,' ,''',' ,'6,' ,'$,' ,'/
        .BYTE   ' ,'-,' ,'2,' ,'!,' ,'^,' ,'7,' ,'1,' ,'(,' ,'_
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
        .BYTE   ' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,' ,'
;
;       ASCII to XXX
;
ASCBDT:
        .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE   11,65, 1,13,55, 1,27,51
        .BYTE   75,23,11,43,15,61,17,57
        .BYTE   33,73,63,41,25,3,53,71,31,07      ; NUMBERS 0-9
        .BYTE   35, 1, 5,37,21,47,45
        .BYTE   61,47,35,45,41,55,27,13           ; LETTERS
        .BYTE   31,65,75,23,17,15, 7,33
        .BYTE   73,25,51, 3,71,37,63,57
        .BYTE   53,43
        .BYTE    1, 1, 1,67,77, 1
        .BYTE   61,47,35,45,41,55,27,13           ; LETTERS
        .BYTE   31,65,75,23,17,15, 7,33
        .BYTE   73,25,51, 3,71,37,63,57
        .BYTE   53,43
        .BYTE    1, 1, 1, 1, 1
        .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ; DO IT AGAIN
        .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE   11,65, 1,13,55, 1,27,51
        .BYTE   75,23,11,43,15,61,17,57
        .BYTE   33,73,63,41,25,3,53,71,31,07      ; NUMBERS 0-9
        .BYTE   35, 1, 5,37,21,47,45
        .BYTE   61,47,35,45,41,55,27,13           ; LETTERS
        .BYTE   31,65,75,23,17,15, 7,33
        .BYTE   73,25,51, 3,71,37,63,57
        .BYTE   53,43
```

```
      .BYTE    1, 1, 1,67,77, 1
      .BYTE   61,47,35,45,41,55,27,13          ; LETTERS
      .BYTE   31,65,75,23,17,15, 7,33
      .BYTE   73,25,51, 3,71,37,63,57
      .BYTE   53,43
      .BYTE    1, 1, 1, 1, 1
;
      .END
```

Don't try to figure out what I'm translating: I can assure you, you don't have any tapes like this.

No prefix files or libraries are required to assemble the translation routine. When assembled, it may be inserted into [1,24]MTA.OLB, where the other ACP modules are.

Next, the file [1,24]MTABLD.ODL must be modified to add the user written routine. Although the appendix states this, it does not say how it should be done: this is a method which works.

Locate the line which looks like this:

```
TRANSL: .FCTR   TRANS-*SY:[1,24]MTA/LB:MTRANS:MEBCDC
```

and replace it with this:

```
TRANSL: .FCTR   TRANS-*SY:[1,24]MTA/LB:MTRANS-*(EBC,UT1)

EBC:    .FCTR   SY:[1,24]MTA/LB:MEBCDC
UT1:    .FCTR   SY:[1,24]MTA/LB:USER1
```

I have shown only one routine, but all three may be added in like manner.

Next the command file [1,24]MTABLD.CMD must be edited: this is very simple, and here the comments in the file tell one exactly what to do. The ACP may now be rebuilt using the MTABLD.CMD task builder command file: note that this will also rebuild the utility MAG, which is not really required.

Mounting a tape with the translation is simple, noting that the switch is /TR=UT1 (or UT2, UT3). For some reason, the /TR switch is not included in the HELP files, so the following SLP file may be of use: it is applied to [1,2]MCRMOU.HLP.

```
-73
       /TR=translation table
-109
 (T)   /TR=translation table   Translate an unlabled tape.
-196
3 TR
   The format of the TR keyword is:

 (T)   /TR=option

   The TR keyword specifies the kind of character translation used
   in read and write operations on an unlabled magnetic tape volume,
   and is normally used with the /NOLAB switch.
   The options are:
```



```
   OPTION    ACTION

   NONE      No translation takes place (the default).

   EBCDIC    Translation between the EBCDIC and ASCII character sets
             takes place. Note that this translation is not symmetrical.

   UT1       Translation between the XXX and ASCII character sets
             takes place.

   UT2       Not implemented.

   UT3       Not implemented.

/
```

You will want to edit it first, to identify the type of tape to be used with each of the user translation switches.

UNIBUS Mapping in RSX-11M

Ken L. Johnson

Monsanto
800 N. Lindbergh
St. Louis, Missouri, 63167

The UNIBUS mapping hardware converts 18 bit UNIBUS addresses to 22 bit physical memory addresses. The UNIBUS map is present on UNIBUS PDP-11s with 22 bit memory addressing; the 11/24, 11/44, and 11/70. The UNIBUS map allows UNIBUS devices to access the 22 bit address space of physical memory with only 18 bits of UNIBUS address space. This is similiar to the memory management hardware which converts 16 bit CPU virtual addresses to 22 bit physical addresses. The UNIBUS map provides no protection or access control function.

There are 31 UNIBUS mapping registers (UMRs), begining at address 17770200. The contents of the UMRs determine the UNIBUS to memory mapping, just as PARs determine CPU virtual to physical memory mapping. Each UMR is two words. The low word is the low 16 bits and the high word is the high 6 bits of a 22 bit base address in physical memory. The UNIBUS map is enabled when bit 20 is set in MMR3.

The high 5 bits of the UNIBUS address determine which UMR is used for the mapping. The remaining 13 bits are an offset added to the base address in the UMR. The correspondence between UNIBUS addresses and UMRs is shown below. Each UMR maps 4K words.

UMR	1	2	...	9	...	31
Starting Address	000000	020000	...	200000	...	740000

The range 760000-777777, which would correspond to a 32nd UMR, is used for the addresses of UNIBUS device registers.

Suppose a DMA device is to transfer a 1 K word block to memory with starting physical address 1450100 using UMR 9. The table above shows that 200000 selects UMR 9 with an offset of 0. Suppose that 200000 is loaded into the address register of the device. Then 1450100 should be loaded into UMR 9, 50100 in the low word and 6 in the high word.

UMRs are referenced in four modules in the RSX-11M V3.2 executive; INITL, CRASH, SYSCM, and IOSUB.

INITL is the initialization routine for the exec. UMRs 1 through 5 are set up in INITL to map the first 20K of physical memory.

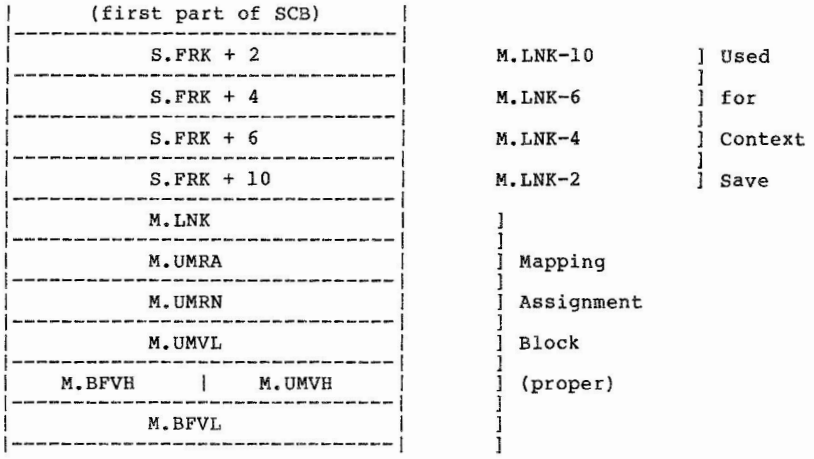
The CRASH module dumps memory to a scratch device when a crash dump is taken. CRASH saves the contents of the UMR's on the crash stack, and they are printed on the first page of a crash dump listing.

SYSCM is the system common area. There are five words in SYSCM needed for UMR usage.

$UMRHD	one word list head for mapping assignment blocks
$UMRHD+2	address of the first assigned UMR (always UMR 1)
$UMRHD+4	N$$UMR = 4 * (number of statically allocated UMRs)
$UMRWT	two word list head for UMR wait queue.

The number of statically allocated UMRs is determined at SYSGEN. The first five UMRs are always allocated. Two more are allocated if the full duplex terminal driver is generated to support a DH-11 multiplexor.

The structure associated with UMR usage is the Mapping Assignment Block (MAB). The MAB proper makes up the last 6 words of an SCB for most 22 bit NPR drivers, immediately following the fork block. Part of the driver's context is saved in a four word context save area if the driver must wait for UMRs. The executive routines which use MABs assume that the four words from the fork block and the MAB are contiguous. A typical MAB as it would appear at the end of an SCB is shown below.



M.LNK	points to M.LNK+2 of next MAB in list
M.UMRA	address of first UMR assigned to this MAB
M.UMRN	4 * (number of UMRs assigned to this MAB
M.UMVL, M.UMVH	UNIBUS address to load in device address register
M.BFVL, M.BFVH	physical address of buffer to be loaded in UMRs

A user-written driver may supply an MAB not in the SCB, but it must be a ten-word block. Therefore "MAB" will refer to this 10 word structure, whether in the SCB or not.

The exec uses two linked lists of MABs. The first, linked from $UMRHD, indicates the current allocation of UMRs. The second, linked from $UMRWT, holds the saved contexts of drivers waiting for UMRs. This is similar in concept to the fork list.

The module IOSUB contains six routines which manipulate these two lists to allocate and deallocate UMRs.

$STMAP (alternate entry point $STMP1) This routine is called by drivers to set up the MAB and allocate the UMRs. $STMAP sets up M.UMRN from U.CNT, and M.BFVH and M.BFVL from U.BUF in the UCB, then calls $ASUMR to allocate the UMRs. If the allocation fails, $STMAP saves R0, R5, and the return address to the driver in the fork block and calls $WTUMR. On return from $WTUMR, restore from the fork block and branch back to call $ASUMR. The alternate entry point is used when specifying an MAB other than the one in the SCB.

$MPUBM (alternate entry point $MPUB1) This routine is called by drivers to actually load the UMRs. $MPUBM is called after $STMAP. $MPUBM uses M.UMRA and M.UMRN to determine which UMRs to load, with the contents

determined from M.BFVH and M.BFVL. $MPUB1 is used to specify an alternate MAB.

$ASUMR This routine is called by $STMAP to allocate the UMRs. $ASUMR finds the first contiguous set of UMRs which satisfy the request and links the MAB into $UMRHD list. M.UMRA, M.UMVL, and M.UMVH are set up by $ASUMR.

$DEUMR This routine unlinks an MAB from the $UMRHD list. $IODON calls this routine to deallocate UMRs assigned to the MAB in the current SCB. Drivers which need to keep UMR's for longer times must therefore use an MAB somewhere else.

$WTUMR This routine is called by $STMAP to wait for UMRs to become available. $WTUMR saves R1, R2, R3, R4, and the return address to $STMAP in the MAB beginning at M.UMRA. $WTUMR then executes a return to the driver's caller. Thus drivers which call $STMAP cannot use the stack for storage.

$DQUMR The address of this routine is pushed on the stack in $IODON. On entry, $DQUMR does a coroutine call back to its caller. Thus when the driver executes a RETURN after calling $IODON, control is transferred to $DQUMR. $DQUMR picks up the first entry in the UMR wait queue. $DQUMR restores R1-R4 from the MAB, and jumps back to the routine which called $WTUMR.

Reflections on Datatreive-11 Version 2.4

John P. Hoekstra

Macatawa Computer Services
1 West Fifth Street
Holland, Michigan 49423

We have just recently (within the last month as of this writing) installed Datatrieve version 2.4 on our system. The following is a compilation of our experiences to date.

Our company is a small systems house specializing in industrial systems design, particularly machine and process control, assembly line automation, and automated test equipment. We do our development on a PDP-11/24 with 256 Kwords of memory and three 10Mb RL02 disks running RSX-11M V4.0.

One of our current projects involves an automated test line that is designed with seven PDP-11s in a DECnet network. The PDP-11s are running a combination of RSX-11M and RSX-11S. The software is written primarily in FORTRAN-77 with some MACRO-11. The data from each of the tests is collected on a central data base machine that serves only to update the data base and generate reports for our client. Our client is relatively new to using automated test equipment and will most certainly change their mind almost weekly about what reports will be useful. The

client is located about an hours drive away from our offices (too short to fly, too long to drive all the time). Our criteria for selecting software to generate reports was the following:

1. It must be compatable with the FORTRAN-77 collection routines.
2. It must be easy to generate new reports.
3. It must be easy to modify old reports.
4. It must be something the client can eventually use themselves.
5. It must be supported software.

The last requirement constrained us to Digital software, the middle three pointed to some kind of query language, and the first one turned out to be no restriction. The only DEC product available on RSX that fits these requirements is Datatrieve. We made a telephone call to our local DEC software person to get his recommendation and then proceeded to purchase Datatrieve for ourselves and a license for our client.

Datatrieve is a 'customer installed' product which usually indicates that it is fairly reasonable to install. We found this to be true. The command file supplied in the kit is rather comprehensive, even allowing you to add manual patches during the installation process. One thing that I would like to have clarified is whether or not it is possible to install Datatrieve using a resident RMS library. I was unable to discover anything about this from the manuals or any files in the distribution kit.

There are three manuals supplied with Datatrieve, a tutorial primer manual, a user manual, and an installation manual. I decided to play 'dumb' and not read any manual before attempting to use Datatrieve. I started playing based on what I had read in the Software Product Description, a lecture I had once attended, and the PDP-11 Software Handbook. I must also state that I had no prior experience with any kind of data base system and no experience with COBOL or other 'business' type languages.

I started by typing 'HELP' just like I was instructed to by the startup message of Datatrieve. I discovered guide mode and started using the 'domains' (data files) supplied with Datatrieve. Guide mode is an interactive mode where you are prompted for all command input and you may enter a '?' at any time to get help based on where in the command line you are. It is something similar to the TOPS-20 command language for those familiar with TOPS-20. Once out of guide mode I discovered that you are required to enter all characters of every command or use predefined abbreviations. As a DCL user who is used to abbreviating all commands to the least number of unambiguous characters this was tiresome at times.

Another feature that took some getting used to was the prompting and error messages. Prompting messages are generated when the syntax of a command is understood but a parameter was missing and take the form of: '[Looking for ...]'. Once I got the hang of treating these as requests for further information instead of error messages I was better off. If you re-enter the command line it generates an error informing you that it was only looking for one word. I am used to reading error messages and going for the manual to discover what the errors mean and what the possible cause is. In Datatrieve there is no further explanation of any error messages because the error messages are very explicit. For instance you might be told that 'the file is open by another user'.

I did eventually read both the user guide and the Datatrieve primer and found them extremely well written. As a further test I gave the primer to one of our technicians who has very limited computer experience and asked him to play. In about a half hour he was doing fairly complicated data base retrieval operations. It is the consensus of the people that have played with Datatrieve so far that, while the manuals have all sorts of examples, the only way to be sure of what Datatrieve will do is to try it since you will never find the exact example in the manual.

My first application was to define existing data files as Datatrieve domains. We had just finished writing a new purchase order program using FORTRAN-77 and indexed files and decided that this would be a good test case. The data in the purchase order system is split between five files so each file needed to be described as a Datatrieve domain and then a 'view' could be constructed of all five files that would have as records all the information in a single purchase order.

I first attempted to use ADT which is the application design tool supplied with Datatrieve. I found out that it insists on representing numbers as character strings while the FORTRAN generated files used the binary representation. This was no big problem since ADT generates indirect command files which can be easily edited to add the 'USAGE IS INTEGER' clause that causes Datatrieve to treat the data as binary. The next problem came when I attempted to define a record that had a non word aligned integer variable. After much searching in the user's guide index I found, in an appendix, the magic clause that told Datatrieve that I wanted to byte align integers. The nice thing about using ADT to define the files is that it delayed my having to learn the COBOL-like definition language. I did eventually learn it and found that it is not so bad after all.

Once all the files were defined as domains and the system was defined as a view we could do many useful operations on our existing data. The first thing that I did was to generate the same report that our FORTRAN system did. The results were: FORTRAN 74 statements, and Datatrieve 2 statements. It is nice to be able to field requests such as 'Did I order any part number XYZ capacitors in the last month?' by turning to the terminal and typing 'Print po with date > 1-aug-82 and description containing "XYZ"' and then waiting for the computer to find any.

Later I designed a 'domain' containing production test results using Datatrieve exclusively. This went very well and I discovered another useful feature of Datatrieve. Since specifications are constantly changing, the data file format changes. In Datatrieve records are defined using field names. If a new field needs to be added the old file can be copied into the new file with a single Datatrieve statement. By keeping all the field, record, and domain names the same there will not be any difference to any user or program. I intend to write some more complicated procedures (Datatrieve programs) to support this new application but have not yet done so.

My overall observations are that Datatrieve is very easy to learn and use. The on-line help is very good although not comprehensive and the manuals are very readable. The really impressive thing to me was that if you can phrase a request in Datatrieve format correctly you will almost always get the desired result.

There are some drawbacks, of course, and we were aware of most of them before we bought Datatrieve so they were not suprises. There is an internal Datatrieve editor that is syntactically the same as EDT that can be used to edit procedures (Datatrieve programs) and data descriptions. However the editor is a very limited subset of EDT which is frustrating since we use EDT exclusively and are used to using all of the features. One particularly annoying feature is that you cannot move backwards from a line, you must go back to the beginning of the buffer and move forwards to the previous line. I have found it convenient at times to 'extract' a definition from Datatrieve into a file and use EDT to edit it and then read the definition back into Datatrieve. I understand that this is not true on the VAX implementation of Datatrieve and that there may also be a way to use EDT in Datatrieve-11 (let me know someone).

The restriction that I was warned about initially was that there would be two types of reporting applications: those that are easy to write using Datatrieve, and those that are impossible to use Datatrieve for. I have found that Datatrieve works best for reports that are 'row and column' reports consisting of headings at the top of the page and data that fits in the columns under the headings. I understand that this is also less true of the VAX implementation of Datatrieve that can use plots and graphs.

In working with my purchase order 'view' that requires five indexed files open at once I find that it is relatively easy to exhaust the Datatrieve pool of space used for sorts and the like. I presume that this is also not a problem in the VAX implementation.

Datatrieve allows entry and modification of records from within Datatrieve. This process gets a bit lengthy since Datatrieve prompts for each piece of data by typing: 'Enter field-name:'. This is acceptable for an occasional entry but could get tiring if you have a lot of data to enter or are on a slow terminal. Once again the VAX implementation gets around this by allowing FMS (Forms Management System) video screens to be defined for data entry.

If you have an application where the file formats and the report formats are likely to change often or you need to have unsophisticated computer users access an existing data base for one-time reports and wish to have them do it themselves Datatrieve might be worth looking into.

Installing Mousetraps in RSX-11M

Carl T. Mickelson

Goodyear Aerospace Corp
Akron, Ohio 44315

In tracking down the problem causing the corruption of location 4 described in a companion article (see Hints and Things), it became necessary to develop a technique for installing Executive patches without re-building all privileged tasks on our system. This article documents how to make room for and install the mousetrap patches to monitor address 4 on all context switches caused by directive requests, interrupts, and returns to user mode. The basic monitoring idea implemented here was suggested by Ralph Stamerjohn and Ken Johnson in their seminar handbook on Crash Dump Analysis.

Our system environment includes a PDP-11/34a, with 256Kb of memory, running RSX-11M, V3.2, Autopatch E, mapped, multiuser. In order to install executive patches one needs space in memory that 1) is accessible in Executive context at all times, and 2) will not change, destroying the patches. The only area of memory outside of the vectors, stack, and Exec code itself that satisfies requirement 1) is POOL. However, requirement 2) is not guaranteed unless some other steps are taken first. Examining the structure of the RSX11M.TSK file and its map, observe that the initialization code in module INITL is executed once and deallocated back to POOL, releasing the area occupied by INITL to be used as scratch space. If there was a way to adjust the size of the memory that INITL deallocates, then the first part of POOL, starting at $POOL would be available for Executive patches. The only restriction is that the patches have to be made to RSX11M.SYS after a virgin system image has been SAV'ed and made bootable.

Following are the steps necessary to make 400 bytes of space available for Exec patches:

```
SET /UIC=[1,54]
PIP RSX11M.SYS/NV/CO/BL:498.=RSX11M.TSK
VMR @SYSVMR
```

Locate in INITL.LST the instruction

```
MOV     $SYBEG-$POOL,R1
```

and note its relocatable address (RA). Find the base address (BA) of INITL in RSX11M.MAP. Then using ZAP, change the word in RSX11M.SYS that will reside at memory address BA + RA + 2. By reducing this word by 400 bytes, we will create a 400 byte patch area at the bottom of POOL when the exec is first booted and saved to disc.

Let BA = 1000(8) * BN + BY, where BN is a disc block number and BY is a byte in block. Then:

```
RUN $ZAP
ZAP>RSX11M.SYS/ABS <cr>
_BN+3:BY+RA/ <cr>
_<nnnnnn> 012701/ <cr>          ;This is the MOV #x,R1 opcode
_<nnnnnn>   mmmm/ mmmm-400 <cr> ;Reduce size of deallocation
_X <cr>                         ;Exit ZAP
```

Now boot the new executive:

```
BOO RSX11M
  .
  .
  .
>TIM <date and time>            ;Set time and date
>SAV /WB                        ;Save with a boot block
```

The system now has 400 bytes of space at the bottom of POOL available for patches to the Executive. This space will be non-zero (it used to be part of INITL), but fixed (it is not a part of POOL).

Now we turn our attention to deciding when the location 4 monitoring should be be done, and what patches should be installed. From Stamerjohn's handbook, the module SYSXT contains the routines that perform context switching to and from executive mode for directives and interrupts.

To monitor location 4, in the $DIRSV routine, the instructions

```
004715      CALL    (R5)
000541      BR      $DIRXT
```

can be replaced with

```
000137      JMP     @#$POOL
$POOL
```

In the $INTSV routine, the instruction

```
042567      BIC     (R5)+,PS
PS offset
```

can be replaced with

```
004737      JSR     PC,@#$POOL+20
$POOL+20
```

In the $DIRXT routine, the instructions

```
012604      MOV     (SP)+,R4
012605      MOV     (SP)+,R5
```

can be replaced with

```
000137      JMP     @#$POOL+40
$POOL+40
```

These enabling patches cause mousetrap routines in POOL to be executed for every directive call and interrupt, and before return to user mode. Next the actual mousetrap routines are designed for installation in POOL.

```
$POOL+00    032737      BIT     #1,@#4
     +02    000001
     +04    000004
     +06    001401      BEQ     1$
     +10    000004      IOT
     +12    004715 1$:  JSR     PC,(R5)
     +14    000137      JMP     @#$DIRXT
     +16    $DIRXT

     +20    042537      BIC     (R5)+,@#PS
     +22    177776
     +24    032737      BIT     #1,@#4
     +26    000001
     +30    000004
     +32    001401      BEQ     2$
     +34    000004      IOT
     +36    000207 2$:  RTS     PC

     +40    012604      MOV     (SP)+,R4
     +42    012605      MOV     (SP)+,R5
     +44    032737      BIT     #1,@#4
     +46    000001
     +50    000004
     +52    001401      BEQ     3$
     +54    000004      IOT
     +56    000002 3$:  RTI
```


Using ZAP, the patches shown above can be installed in RSX11M.SYS startins at $POOL. These patches are then resident in memory each time the system is bootstrapped. The enabling patches shown above should not be installed permanently with ZAP. In case of a ZAP patching error, you will have corrupted the only bootable exec on the disc. The error will be in such a basic area of the exec that it will be impossible to complete booting the system.

To start monitoring the content of location 4, first re-boot the system to load the mousetrap code into memory. Next, halt the processor, and manually insert the enabling patches into memory using the programmer's console. Then continue the processor, and if location 4 is found to be odd at any context switch, the system will IO trap to XDT. If the PC is at $POOL+12, the current user task (which must be privileged) corrupted location 4. A PC at $POOL+36 indicates location 4 was odd at an interrupt, while $POOL+56 indicates location 4 was corrupted by the exec, a device driver, or the current privileged program while running in system state.

RX02/RX03 Problems in RSX-11M V4.0

Greg Liverman

MPI fuer Quantenoptik
8046 Garching bei Muenchen
West Germany

I have come across two rather serious problems with the RSX-11M V4.0 DY: driver. The first is that the driver no longer supports double sided drives (the so called RX03) and the second is that the driver manages to cause the executive to become corrupted if an error occurs.

Before going any further, I would like to make clear that I am the user of a second source, DEC compatible floppy disk system containing double sided drives. This system is a DSD-880 floppy/Winchester system manufactured by Data Systems Design of San Jose, California. We have never had any compatibility problems, or any other problems, for that matter, with these systems in the past. I am mentioning this because the first problem definitely effects only users of second source hardware and the second problem might also only effect users of non-DEC hardware.

Problem 1 - Missing Double Sided Support

The DY: driver no longer supports double sided opperation. Support for double sided floppy drives has been available in RSX-11M V3.2 for some time. Now its gone! For installations like ours that rely heavily on double sided floppies, this is nothing short of a major catastrophe. Our users, for example, are responsible for keeping their own personal files backed up. We have found the RX03 (double sided, double density, 1Mb capacity) floppies to be ideal for that prupose. On our data collection machines, we have only a floppy drive and a Winchester.

The cause of the difficulty is was easily discovered. There is a comment near the beginning of the floppy driver:

P. J. CARR 29-APR-80

PJC006 -- REMOVE SECOND-SIDE SUPPORT

The solution is also very straightforward. You take the patched V3.2 DYDRV.MAC and compare it with the new, V4.0 DYDRV.MAC and then edit the V4.0 file to contain the second side support code. This cannot be done with a CMP and SLP operation because of all the other changes that have been made to the driver in V4.0 (error logging and reordering of labels, for example). After the V4.0 driver is edited, a CMP with the old V4.0 driver will give you a SLP correction file.

How does the new driver work? I have been using it for about a month without any problems. The DEV command reports "TYPE=RX03" just like it used to when a double sided floppy is put in the drive. BAD and INI handle double sided floppies without problem. FLX, however, reports the media as illegal (I have forgotten the exact message) if an RX03 floppy is used. This is rather serious for people who use RX03 floppies on an RT-11 system and would like to use them to transfer data to and from a RSX-11M system. Luckily, the FLX from V3.2 runs quite nicely under V4.0 and it allows RX03 floppies. So don't throw your V3.2 FLX away! I have not yet tried other programs like FMT or IOX. It could be that these might balk at RX03 floppies also.

This whole problem has made me rather upset with our mutual friend, Digital Equipment Corporation. The reinstallation of second sided support in the driver was not very difficult and I can maintain that myself without too much trouble. This is because I have the sources for the driver. However, the FLX problem is quite a bit more difficult to handle. If I want to use double sided RT-11 floppies with FLX I must either use a stagnant V3.2 FLX or I must find a way to patch FLX. As far as I know, you need the sources to write an object code patch and I don't have them! I can, of course, turn to (on?) my floppy disk manufacturer and demand a patch for FLX. I would probably get it, too. But then I must demand that patch every time DEC makes a patch to FLX. All this demanding and patching sounds to me like a lot of work. Is DEC trying to tell us that our RX03 hardware is too much trouble to use with RSX-11M? Are they trying to tell us to get rid of it? I don't like the implication of that line of thought at all!

Problem 2 - Executive Corruption After Errors

I have observed this problem on my system using <u>single sided</u> floppies. However, as will be clear later, the problem could be due to the fact that I don't have DEC hardware.

The problem is easily reproduced as follows:

1. Log on into a priviliged account.

2. Insert a double density floppy and mount it with the "/OVR" switch. The floppy should, of course, be FILES-11. It should mount.

3. Dismount the floppy and remove it from the drive.

4. Insert a single density, FILES-11 floppy and mount it with the same command. The floppy should make a lot of noise loading and unloading the heads. It should mount.

5. Dismount the floppy and remove it from the drive.

6. Replace the double density floppy in the drive and attempt to mount it again with the "/OVR" switch. The floppy should make a lot of noise loading and unloading the heads. You should get the error "Incorrect volume label" from MOU.

7. Attempt to mount another device (DL:, DM:, etc.) with the "/OVR" switch. You will always get the error "Incorrect volume label". Specifying the correct volume label will mount any device, however.

This problem was rather difficult to analyze. I first observed the problem in the way described above and it led me to think that MOU was in error. MOU does a read logical block on block 1 of every device (foreign or otherwise) it tries to mount in order to make sure the volume is loaded and up to speed. That's where the floppy clicks a lot. I patched the MOU source file MOUSUB.MAC such that a sense characteristics QIO was done before the read if the device was a floppy (DY:). I then rebuilt MOU and the problem was solved. Due to other, more pressing problems (they keep cropping up!), I gave this problem no more thought and declared it solved.

When I started to write this article, I began to think about the problem in a more relaxed manner. I knew that the only difference between those two floppies was the density and that the floppy driver was supposed to recover a simple density error in one retry. But from all the noise that was being made it was quite obvious that more that one retry was being made.

I wrote a little test program that just read one block from the floppy (IO.RLB) and reported the I/O status word values. Then I started to play around with it to see if I could get some more information. One time, after having recreated the problem as described above with my test program, the system hung. I was using ODT at the time and the hang had occurred immediately after I tried to open a nonexistent address. That meant the hang had occurred during or immediately after a trap. I therefore suspected that the vector area was corrupted. I halted the computer and examined the vector area. Locations 4 through 12 were shot. Now I was getting somewhere! The next thing to find out was <u>how</u>!

An examination of the corrupted memory locations told me that they had been put there by a DMA transfer from the floppy hardware. After every error, the driver allocates 6 words of core, loads two words itself with error information and then causes the hardware to DMA four words of error information into the other four words. The four corrupted vectors were loaded with what looked suspiciously like this error information. I first checked the hardware, but it was O.K. The next move was to figure out how the hardware was getting an address of 4 as the memory address for a DMA!

I read and re-read the driver code trying to find an error. I even checked all the calls to the system subroutines, those that allocate core blocks, for example, but it was all clean. So I threw a couple of HALT statements into the driver code at choice places, rebuilt the driver, loaded it and started to play. It didn't take long to find out where the problem lay. The DMA procedure for RX02 hardware is that first a function code is loaded into the control and status register (CSR), then the program must wait until the transfer bit (TR bit) is set in the CSR and then the program loads a memory address into the data

buffer resgister (DBR). The driver was simply not witing long enough for the TR bit to become set, was never giving the memory address to the hardware, was never completing the DMA setup procedure and therefore royally messing up everything! The driver checks the TR bit in a loop and this loop has a counter on it - the only check_the_transfer_bit loop in the whole driver with a counter! I never completely figured out how, where or why, but at some unknown time later, the DMA procedure was being completed to the satisfaction of my hardware and it DMAed four words of data into memory. It just turned out that those four words of memory happened to be very important!

The fix was very simple. I just raised the loop counter to a higher number. I have had no problem since. This problem, as I mentioned previously, might only effect my hardware. It could be that DEC hardware or hardware from other manufacturers responds with the TRANSFER bit set much more quickly and that this problem never occurs. However, I really don't see why a counter is necessary here at all. If, for some reason the transfer bit never becomes set, the driver will hang. O.K. But is that worse than having the executive corrupted by a wild DMA transfer? I'd rather have the hang, thank you.

And now - The Patch

The following is the patch to be applied to a copy of the file DYDRV.MAC from the RSX-11M V4.0 distribution kit. If you don't want the double sided support, then delete all the lines below that look like or come after a line of the form:

```
        -nn,nn,/; MGL001/
```

and before the next line that begins with a "-". The lines beginning with a "-" mark the beginning of a SLP command. All the lines up to, but not including, the next line beginning with a "-" are part of that SLP command. The SLP commands marked with "MGL001" are for double sided support.

```
[11,10]DYDRV.MAC;2/-BF/AU:72.=[11,10]DYDRV.MAC;1
-77,,/; MGL001/
;       M.G. Liverman   26-AUG-82
;
;               MGL001 -- Replace double sided support
;
-,,/; MGL002/
;       M.G. Liverman   26-AUG-82
;
;               MGL002 -- Fix error logging problems and
;                         function code comparisons
;
-123,,/; MGL001/
SSIDED  =       20000           ; Second side indicator bit
-132,,/; MGL001/
SINSIN  =       494.            ; Single sided and single density
SINDOU  =       988.            ; Single sided and double density
DOUDOU  =       1976.           ; Double sided and double density
-146,,/; MGL001/
DSIDED  =       1000            ; Head select bit
-155,,/; MGL001/
SIDES   =       2               ; Double sided media
```



```
-401,401,/; MGL001/
220$:   BIT     #SSIDED,U.CW2(R5) ; Is it second side operation?
        BEQ     225$              ; If EQ no
        BIS     #DSIDED,U.BUF(R5) ; Use second side
225$:                             ; Ref. label
-528,528,/; MGL002/
        CMPB    #IO.SEC/256.,I.FCN+1(R1) ; IO.SEC or IO.SMD func's
-537,537,/; MGL001/
420$:   BIC     #SILO!SCHAR!SSIDED!ERR1,U.CW2(R5) ; Clear all but density bits
-562,563,/; MGL001/
440$:   BIC     #SCHAR!DEN!DSIDED,U.CW2(R5) ; Clear flags
        MOV     #SINSIN,U.CW3(R5) ; Assume single density, single sided
-568,568,/; MGL001/
450$:   BIT     #SIDES,I.PRM+6(R1) ; Is it double sided?
        BEQ     455$              ; If EQ no
        ASL     U.CW3(R5)         ; Double the mAximum LBN's
        BIS     #DSIDED,U.CW2(R5) ; Set the double sided bit
455$:   MOV     #IS.SUC&377,R0    ; Set success
-588,588,/; MGL001/
        MOV     #SINSIN,U.CW3(R5) ; Assume single density
-660,660,/; MGL001/
        MOV     #SINSIN,U.CW3(R5) ; Change max LBN's to single density,
                                  ; Single sided
-664,665,/; MGL001/
        MOV     #SINDOU,U.CW3(R5) ; Change max LBN's to double density,
                                  ; Single sided
560$:   BIT     #SIDES,I.PRM+6(R1) ; Is it double sided?
        BEQ     565$              ; If EQ no
        BIS     #DSIDED,U.CW2(R5) ; Set double sided bit
        ASL     U.CW3(R5)         ; Double max LBN's
565$:   MOV     R0,I.PRM+10(R1)   ; Store logical sector number
-725,,/; MGL002/
        BR      80$               ; Leave - allocation failure
-733,733,/; MGL002/
30$:    MOV     #100,R3           ; Set up loop count
-750,751,/; MGL002/
65$:    BITB    #DONE,(R2)        ; Done?
        BEQ     65$               ; If EQ no
-792,796,/; MGL001/
        CMP     R0,#76.           ; Is it second side?
        BLT     30$               ; If LT no
        BITB    #IO.RPB&377,I.FCN(R3) ; Physical block function?
        BEQ     23$               ; If EQ no, it's a logical block
        CMP     #76.,R0           ; Yes
        BEQ     30$               ; If EQ allow access to #76.
        SUB     #77.,R0           ; Change sides - phyblk access
        BR      27$               ; Change read heads
23$:    SUB     #76.,R0           ; Adjust for second side
27$:    BIT     #DSIDED,U.CW2(R5) ; Two sided media?
        BEQ     60$               ; If EQ no, bad block
        BIS     #SSIDED,U.CW2(R5) ; Set head 1 select bit
/
```

The patch must be applied with SLP and then DYDRV must be rebuilt according to the directions in the Release Notes. This patch must be reapplied after every DEC patch. It must be reapplied manually (with EDI, EDT, TECO or your favorite editor) because this patch will not work on a patched file. By the same token, DEC patches won't work on a patched file, either, so you can't get around it by running this patch first. I would hope that DEC would include all of the above in the Software Dispatch and the next Autopatch kit (I have submitted an SPR), but if it turns out that real DEC RX02's don't have the problem with the TR bit taking a long time to become set, it could be that only one line of the stuff above (CMPB #IO.SEC/256.,I.FCN+1(R1)) could find its way into the Software Dispatch.

RMDEMO I/O Display Page for RSX-11M V4.0

Jorma Koski

SYSPLAN
Uudenmaankatu 29A6
SF-00120 Helsinki 12
Finland

There is a display page in RMD which will work allso on M with out PLUS. It is called I/O page and applying following SLP correction files You may allso use it. Note that there is a special feature for disk devices named SM: it is a user written driver which keeps its own word count and calculates blocks per I/O just for RMD. The driver doesnt use the DV.MSD bit in U.CW1 so it increments U.IOC and U.IOC+2 it self.

---- Contents of IOPAGE.COR ----

```
[14,10]IOPAGE.MAC/AU:72,=[14,10]IOPAGE.MAC;1/-BF
-12
; MODIFIED BY:
;
; JTK03 82-07-28          Jorma Koski / SYSPLAN ky
;                         Correct M v4.0 code using U.IOC
;                         Add words per second for SMDRV
;                         Change cylinders per I/O to blocks for SMDRV
%
-44,44,/;JTK03/
        .IF NDF R$$MPL
        UCBDF$  ,,,SYSDF        ; 'SYSDF' to set U.IOC etc.
S.ST2   = S.PRI                 ; No S.ST2 in M v4.0 ..
S2.LOG  = 4                     ; But error log bit is this one in S.PRI
S2.OPT  = 340                   ; Set all devices to support cyl. opt.
        .IFF
```



```
        UCBDF$
        .ENDC
-114,118,/;JTK03/
        CMP     @R0,#'SM                ; Is it CDC9762 or CDC9766
        BEQ     2010$                   ; EQ - yes, thats our driver
        MOVB    U.ERSL(R2),ERSL(R1)     ; DEC drivers keep error counts
        MOVB    U.ERSC(R2),ERSC(R1)     ; ..here if they support error
        MOVB    U.ERHL(R2),ERHL(R1)     ; ..logging, but SMDRV does
        MOVB    U.ERHC(R2),ERHC(R1)     ; ..not do it yet...
        CLR     WRDCNT(R1)              ; So DEC devices dont keep
        CLR     WRDCNT+2(R1)            ; ..word counts
        CLR     CYLCNT(R1)              ; ..nor do they set
        CLR     CYLCNT+2(R1)            ; ..block counts
        BR      2015$                   ; Branch over SMDRV special code

2010$:                                  ; SMDRV special code starts here
U.WCNT  = 36                            ; Define this one now and here
        MOV     U.WCNT(R2),WRDCNT+2(R1) ; We keep word counts here
        MOV     U.WCNT+2(R2),WRDCNT(R1) ; ..until otherwise done
        MOVB    WRDCNT(R1),CYLCNT+3(R1) ; Convert words to blocks
        MOVB    WRDCNT+3(R1),CYLCNT+2(R1);..by dividing it by 256.
        MOVB    WRDCNT+1(R1),CYLCNT(R1) ; ..with move-bytes
        CLRB    CYLCNT+1(R1)            ; High byte allways zero
2015$:                                  ; End-of-SMDRV code
-205,205,/;JTK03/
;;;     .IF     DF,R$$MPL               ; Allways take this code with me
-441,441,/;JTK03/
;;;     .ENDC                           ; End-of-allways
/
```

---- Contents of RMDDEF.COR ----

```
[14,10]RMDDEF.MAC/AU:72,=[14,10]RMDDEF.MAC;1/-BF
-59,,/;JTK03/
        .GLOBL  $ABTIM
        .GLOBL  $ERFLA
        .GLOBL  ES.LOG
-95,95,/;JTK03/
-109,110,/;JTK03/
/
```

---- Contents of HLINIT.COR ----

```
[14,10]HLINIT.MAC/AU:72,=[14,10]HLINIT.MAC;1/-BF
-69,77,/;JTK03/
        .ASCIZ  'I - I/O Counts Display'
        .ASCIZ  /M - Memory Display/
        .ASCIZ  /S - System Statistics Display, M-PLUS only/
/
```

---- Contents of IOINIT.COR ----

```
[14,10]IOINIT.MAC/AU:72,=[14,10]IOINIT.MAC;1/-BF
-45,45,/;JTK03/
        .IF DF  R$$MPL
HEAD7:  .ASCIZ  /Cylinders/
        .IFF
HEAD7:  .ASCIZ  /Blocks   /
        .ENDC
/
```

```
        ---- Contents of RMDXCM.COR ----

[14,10]RMDXCM.MAC/AU:72.=[14,10]RMDXCM.MAC;1/-BF
-40,40,/$JTK03/
$MDDEV:.ASCII  /  /             ; DEVICE NAME, FILLED IN AUTOMATICLY
/

        ---- Contents of RMDBLD.COR ----

[1,20]RMDBLD.BLD/-AU=[1,20]RMDBLD.BLD;1/-BF
-50,51
.DATA GBLPAT=RMDEMO:$PGTBL+122:47511
.DATA GBLPAT=RMDEMO:$PGTBL+146:51523
-86,86
.IFF $11MPL .DATA MAPS:   .FCTR  (MDMAPS,HLMAPS,ATMAPS,THMAPS,IOMAPS)
-96,96
.IFF $11MPL .DATA IOMAPS:.FCTR  (GBLIOM,V52IOM,US1IOM,V1HIOM,USQIOM)
-100,100
.IFF $11MPL .DATA COMMON:.FCTR  (MDCOM,HLCOM,ATCOM,THCOM,IOCOM)
-105,105
.IFF $11MPL .DATA PAGES:  .FCTR  (MD,HL,AT,TH,IO)
-111,111
            .DATA IO:     .FCTR  (IOINIT,IOPAGE,IOCMD)
-125,125
            .DATA GBLIOM:.FCTR  '$LI'RMD/LB:GBLIOM
-133,136
            .DATA IOCMD:  .FCTR  '$LI'RMD/LB:IOCMD
            .DATA IOCOM:  .FCTR  '$LI'RMD/LB:IOCOM
            .DATA IOINIT:.FCTR  '$LI'RMD/LB:IOINIT
            .DATA IOPAGE:.FCTR  '$LI'RMD/LB:IOPAGE
-167,167
            .DATA V1HIOM:.FCTR  '$LI'RMD/LB:V1HIOM
-180,180
            .DATA V52IOM:.FCTR  '$LI'RMD/LB:V52IOM
/

        ---- End-Of-Files ----


To apply patches:

SET /UIC=[14,10]
...edit in correction files...
SLP @RMDDEF.COR
SLP @HLINIT.COR
SLP @IOPAGE.COR
SLP @IOINIT.COR
SLP @RMDXCM.COR
SET /UIC=[1,20]         ! NOTE: patching file RMDBLD.BLD
SLP @[14,10]RMDBLD.COR
SET /UIC=[14,24]
```



```
MAC
 IOPAGE=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[14,10]RMDMAC/PA:1,IOPAGE
 Assemble allso HLINIT,IOPAGE,IOINIT,RMDXCM,V1HIOM and V52IOM
SET /UIC=[1,24]
PIP /NV=RMD.OLB         ! Save good old library just in case...
LBR
 RMD/RP=[14,24]RMDDEF,HLINIT,IOPAGE,V1HIOM,V52IOM,IOINIT,RMDXCM
SET /UIC=[200,200]
@SYSGEN3                ! Use this file to create RMD.TSK
```

SPR Survey Results

Ralph W. Stamerjohn
Multi-Tasker Editor

The SPR Survey published in the December/January 1982 issue of the Multi-Tasker resulted in 262 responses, representing 411 readers of the Multi-Tasker and 704 systems: 593 RSX-11M, 27 RSX-11S, 45 RSX-11M Plus, and 39 IAS.

This first thing I noticed in the responses is that I did a poor job writing the survey. The results, especially the eight questions, are very difficult to quantify into numbers. I should have asked very specific questions that lended themselves to computer tabulation.

However, the people responding obviously took a great deal of time to carefully consider the questions and made many valid comments on the Software Maintenance Services. In the sections below, I will try to summarize these responses and quote from the surveys statements which are especially interesting.

1.0 MAJOR RESULTS

I saw two major points made by the responses: an Autopatch mechanism that people trust is critical and people still have problems with the order processing system. Also, a disturbing trend is occuring. Only a very few number of sites are reporting problems to Digital.

The first major point suprised me. Many sites reported they did not use the Software Dispatch for fixing problems. Instead, they used it to research problems and relied on the Autopatch kits to make the actual fixes. Errors in the Dispatch fixes did not seem to bother many people. But errors in Autopatch did. Also, people commented on the need for improvements in how Autopatch is engineered and complete documentation on exactly what problems are being patched.

The order processing system also did poorly in the survey. Comments of sites receiving too few or too many Dispatches, use of wrong addresses, lapses in service between contract renewals were quite common.

Finally, I observed that only a few sites are reporting problems to Digital. A total of 1043 SPR's were reported by the 191 sites who had the service. But 7 sites accounted for 66% of the SPR's (685) and 15% of the sites submitted 88% of all SPR's. Over half (104) have never submitted an SPR.

I find this disturbing. If a problem, especially a minor one, is not reported to Digital, there is a little chance it will never be fixed. And if the few sites using the SPR mechanism give up or move to other systems, it appears such problems will not be reported.

2.0 NUMERICAL RESULTS

The table below summarizes the numerical results from the survey. The numbers should not be taken too literally, because of the small size of the survey responses.

I would like to point out special attention to some of the numbers. The total of 75 SPR's out of 1043 printed in the Software Dispatch is probably smaller than actual. But I do find the relationship of this value to the number of fixes and workarounds users received in response (75 to 382) a reflection of the number of problems reported which are never published in the Dispatch.

Total number of survey responses:	262		
Number of Multi-Taskers represented:	411		
Total number of systems:	705		
RSX-11M	593		
RSX-11S	27		
RSX-11M+	45		
IAS	39		
Total number of sites not using SMS:	71	(27.1% of total response)	
Total systems not covered by any SMS:	127	(18.0% of total systems)	
Total sites covered by some type SMS:	191	(72.9% of total response)	
Through Autopatch Service:	172	(90.0% of SMS)	
Through Telephone Support:	91	(47.6% of SMS)	
Through DECsupport Service:	10	(5.2% of SMS)	
Total SPR's submitted in survey:	1043		
SPR's also submitted with fix:	156	(14.9%)	
SPR's published in Dispatch:	75	(7.2%)	
Breakdown by problem severity:			
Heavy system impact (1):	164	(15.7%)	
Moderate system impact (2):	206	(19.8%)	
Minor system impact (3):	281	(26.9%)	
No significant impact (4):	202	(19.3%)	
Documentation/suggestion (5):	191	(18.3%)	
Breakdown by Digital response:			
Patch which fixed problem:	235	(22.5%)	
Workaround solved problem:	147	(14.1%)	
Fixed in next release:	181	(17.2%)	
Problem is restriction:	103	(9.9%)	
Could not reproduce problem:	42	(4.0%)	
User problem:	45	(4.3%)	
Thank you for SPR:	99	(9.5%)	
Response did not fix problem:	86	(8.2%)	
SPR still pending:	71	(6.8%)	
Other classification:	45	(4.3%)	

3.0 SURVEY QUESTIONS

The following summarizes the response for the eight questions asked on the survey.

3.1 Q1: Sites Which Do Not Use SMS

The 71 sites not using Digital Software Maintenance Services use a wide variety of different means, ranging from ignoring any problems to outside consulting. Other forms of maintenance were in-house experts, updates from the OEM responsible for the system, and reliance on an underground which includes the Multi-Tasker.

3.2 Q2: Problems Getting The Software Dispatch

"Sure, hasn't everybody?"

This quote seems to best reflect the experiences the surveyed sites have had getting the Software Dispatch. Over 120 response noted some form of problem, though many of these were in the past and have sinced been fixed.

The most common problem is simple: people receiving too many dispatches. I suspect much of this is caused between the differences in warranty and subscription services. The warranty period for the operating systems and layered products includes a separate copy of the Software Dispatch. But only operating system subscription services come with the Software Dispatch. Dispatches are not included in maintenance services for layered products.

This problem is only serious in that the unncessary copies Digital must print and mail raises the cost of maintenance services. But about 40 sites noted the more serious problem of not receiving the Software Dispatch for various periods of time and various reasons.

It is hard, from the survey, to pinpoint any single problem that causes mishandling of Dispatches. It seems in part to be related to individual sites purchasing mechanism. The order for Software Maintenance Services is expensive enough that it must pass through many hands and there is plenty of opportunity for the contact name and address to be lost in the paper shuffle. It also

appears that a similar paper shuffle occurs when the order is sent to the local Digital office.

What seems to make the problem worse is that there is not a simple, one-call method to use once a problem occurs. The cover letter in the Dispatches states "Address change requests should be sent to the nearest DIGITAL field office. Include the new address and mailing label from the most recently received publication." There were several comments that matched my experiences that it can take over a year to get a change made in any Dispatch mailings from Massachusetts.

"Sure, hasn't everybody? Currently, we receive a least two and sometimes three copies of the Software Dispatch for RSX-11M/M-Plus and one for a VAX which we don't have."

"Yes, Dispatch routinely sent to wrong person, wrong number of dispatches sent, cases where Dispatch never received. Unable to determine how to correct the problem."

"One year we got 7 (count'em) copies. Next year, one copy about 2 out of every three months. More recently, two copies sent to a wrong (but consistent) mailing address, so we chose to live with this."

"Yes. The usual problem was that the Software Dispatch was addressed to the apparently random person who signed the purchase order. However, there are about 2000 people and a dozen PDP-11's here, and many of the people who sign our larger purchase orders do not remeber who uses the Software Dispatch (and some are not even at the location anymore). We are still missing 6 issues. The latest issues are finally consistently addressed to the right person, after repeated requests for address corrections over a period of 2 years."

"I get three copies, need only one."

"I work for the Federal Government. The government does not sign a contract with Digital until four months after the old one expires so I miss four months and am not sure I will get the back issues or not."

"Being sent to person who is no longer with company. Neither the Software Dispatch or envelope gives any idea where to send to make address corrections."

"Yes. Four month lapse in service at last contract renewal. Had to stop payment to get DEC's attention."

"Every once in a while we get a package marked 'priority mail', containing copies of the Software Dispatches that we already have."

"Yes, but please do not try to correct. I tryed twice already and both times stopped geting any."

"Wrong address. Dispatch mailed to same address as equipment: namely our warehouse rather than office."

"Four months after purchase order accepted, I am yet to receive Dispatch

via regular channels. Local DEC office has been good at bootlegging me copies."

"Yes, even though we have Autopatch too, the Dispatch never arrives. We have to have it sent special every 3-4 months."

"Yes. We ordered RSX-11M V3.2 in July 1980. Through some screwup at DEC we didn't get the software till around 12/80., by which time our 90 day warranty had allegedly expired. They appologized profusely, but would not send any SPR's beyond 11/80."

"Yes, sent to wrong address, even after notification."

"Yes, there is another PDP-11 in another department running RT-11, but somehow our order files were confused or merged so that for a long time they received our dispatches."

"Took over 1 year to receive Dispatch for RSX-11M and took 1 year to stop excess copies."

"Have had gaps in service, probably caused by red-tape in our purchasing department."

"YES! Software dispatches stop coming when Software Maintenance contract is renewed. It takes several phone calls to start them coming again."

"Dispatches went to OEM who held system license. Missed several for this reason."

"Frequently over the years. However, it is much better now (last two years)."

"Yes. It almost takes an 'Act of Congress' to get name change, reduction of the number of copies, and obtaining all published copies."

3.3 Q3: Experiences With SPR Mechanism

People who use SPR's reported general satisfaction with the mechanism. The biggest complaint was the slowness and paper work, but these were recognized as a inherent part of the system and not a major problem with the availablity of Telephone Support. People also reported a wide variety in the service.

Some of the major users of SPR's noted they use the mechanism to report problems they had already fixed and were disappointed Digital did not then make this information generally available.

It was interesting to see that some of the most negative comments came from sites that indicated no SPR's submitted and the really large users had generally good things to say. This would seem to indicate that the SPR mechanism has a reputation it does not deserve.

"I think the quality and turnaround of SPR's has been acceptable. It is difficult to judge without knowing how many SPR's DEC processes, but I

would say I am satisfied with the service."

"SPR mechanism has differed continually in quality. Responses have ranged from timely and correct to late and surly."

"No sense of urgency."

"Good. Response time good."

"The turnaround is usually one week to send back confirmation of receipt, 4 weeks for answer. This is OK I guess. TSC helps somewhat for turnaround."

"They seem to try hard for us."

"Terribly slow - we still have some SPR's outstanding for a year or more, but improving. Because of this we either put in a correction ourselves or find a workaround. It is quicker for us to spend a week or two sorting problems out rather that wait three months for a response."

"In general, the SPR responses I have received have been useful. I have had a few cases where the response has been of no use or received much too late to be of use."

"My SPR was treated as a undocumented restriction! We lost a few man-weeks on this problem.

3.4 Q4: How Effective Is The Software Dispatch?

The Software Dispatch, when people people receive it, seems to be adequate. The most common comment on print quality was a need to distinguish between zero's and the letter "O". It seemed from the survey that people accepted the fact that patches in the Dispatch would have occasional errors, but because Autopatch is the primary patching mechanism, caused no great problem.

"It works. 80% of the patches are easy to apply."

"Reasonable quality, on a rising curve. No recent problems."

"I usually wait for Autopatch, unless the patch will significantly improve system performance or reliability."

"I use the Dispatch as a reference only. I never apply patches manually but always use Autopatch. I like the Software Dispatch though."

"We faithfully apply the Software Dispatch each month. We find bad patches maybe 6 times each year."

"Rarely apply patches, but like to know about problems."

"Dispatch is effective was to find additional data on known problems, but only if they are published!"

"Software Dispatch is effective. I only apply patches after I have received 6 months worth. This is to avoid problems with corrections to previous patches."

"Be sure Software Dispatch is completely legible."

3.5 Q5: How Effective Is The Autopatch Kit?

Autopatch got mixed reviews. Almost everyone who uses Autopatch agreed that it was a great timesaver for patching. There were three common complaints: (1) The documentation on the Autopatch kits is not adequate. You need to know what you are patching and how the patches compare to articles published in the Software Dispatch. (2) The human engineering of the Autopatch kits needs to be greatly improved. The Autopatch command files are too inflexible for the actual user environments and make too many assumptions. (3) The patches and patching mechanism should be consistent for all products.

The feeling you get reading the survey responses is people have high hopes for Autopatch but have been burned. If a few kits appeared in a row with no major problems on them, user acceptance and trust would shoot way up.

"I use Autopatch only as a source of machine readable patches. I have had many problems with the deleted files, new task image that do not work and so on. However, I still find a real time saver over generating patches by hand."

"Yes, sort of. We only apply those patches on Autopatch that (1) have not already been published in the Software Dispatch and (2) they appear to be critical patches. We have too many local mods to the system to allow Autopatch to reck its havoc."

"The Autopatch kits are not consistent in the way patches are done (e.g. SORT-11 verus RSX). Also UIC's are not consistent with proper target UIC's."

"Never use Autopatch command files, but find, after cleaning up, most patches are useful and usually not disastrous."

"O.K. No problems except the tape is always late."

"I have seen errors that if one actualy did the autopatch off the command file they would obviously have seen."

"Rather effective."

"The autopatch is much better than using the dispatches! I now wait for that rather then typing in the patches."

"Problems caused by Autopatches do tend to decrease confidence.

3.6 Q6: How Effective Is Telephone Support?

Like Autopatch, TSC got mixed reveiws. Almost everyone who has used TSC agreed it worked very well when they were starting up. But TSC seems to fail when sites got more experienced and problems became more complex. This is quite understandable. We all realize the difficultly in solving complex problems over the phone. Some of this problem, however, is due to the lack of experience in TSC specialist and the resources available to them.

The major complaint against TSC was its cost. Many surveys asked for some sort of per-call billing. People felt the current cost does not justify the benefits.

"It would not be fair to ask TSC for an immediate answer, but I do appreciate TSC serving as a research tool. I can call up and say I have a problem and they never fail to get back an answer and they are only satisfied when I am satisfied with the answer."

"Telephone Support has not been much help. It seems like after you give them enough information to convince anyone that it is a system problem, they still act like it is a problem with your application, e.g. EDT V2 has a few bugs, but trying to convince them of that gives anyone a headache."

"Would be very effective for a neophyte user, but not much when problem analysis requires looking at a crash dump, for instance. Their structure is such that they couldn't help with anything complicated."

"Generally helpful and friendly...Call-back times are sometimes long, even for emergency/short questions."

"Inadequate depth and breadth of knowledge and experience. The problem is that nobody who knows enough about RSX to be of real help would want to do this as a job."

"All questions haved been answered...We feel this is a good service."

"Telphone support was moderately helpful during warranty period, but not helpful enough to warrant exorbitant monthly charge to continue!"

"Very helpful. Rarely were they unable to help us with our problems. A few individuals wre somewhat disinterested (very few)."

"The people at TSC should have more documentation than we do. It does not help when the TSC specialist is reading the same inadquate manual you are."

3.7 Q7: Suggestions To Improve SMS

The suggestions added up to publishing all known problems with 100% accuracy, having frequent, correct Autopatch kits, and staffing TSC with an army of wizards. In short a perfect system.

The most common suggestion concerned publishing known problems. Sites seemed to want a list of known problems so they (1) can diagnosis quickly whether a problem is in their or Digital software and (2) not duplicate effort reporting problems already known to Digital.

"I would find it very helpful if all SPR's were published in the Dispatch as they are received."

"No, I am impressed with their efficient and courteous service."

"How can you cover the huge array of software products or user-base without the great expense of money necessary on hiring more people? I think I get lots of service for by $500/month."

"Put all known errors, bugs, patches in the Dispatch."

"Publish all SPR's for problems that have working fixes. Summarize other known problems."

"Yes, standardize and document Autopatch and layered product UIC's and form of patches. Publish all SPR's. Provide more feedback on SPR's as as to our quality in filling the things out. Put hardware product descriptions in the back of the Software Dispatch."

"Patches should be more thoughly checked before release. There should be better coordination between the Dispatch and Autopatch. Telephone support should be staffed by people of higher technical level.

"More careful checkout of published patches. Elimination of clerical errors in order processing. Provide a clear statement of what levels of service are available at what price. I am not absolutely sure which level of service we currently have. They keep changing the name!"

3.8 Q8: What Should The RSX/IAS SIG Be Doing?

"I don't know."

This comment occured several times. It also reflects my own personal feelings. Many people commented that the SIG should not do Digital's job. Others wanted the SIG to do more, effectively starting its own Software Maintenance Service and go in competition with Digital.

"Yes. It should be made clear to DEC that software maintence is an increasingly large percentage of DP budgest and it is also a consideration inevaluation of new and replacement systems."

"SIG role should be as a catalyst to improve DEC response to problems in

user community. SIG personnel are not DEC employees and should not be fixing problems for DEC unless that are componsated by DEC. Just keep jabbing them."

"Yes! The SIG should insure that the DEC software support personnel remain responsive to customer needs. DEC is getting extremely large and more difficult to deal with, it should be the SIG's goal to keep DEC responsive and aware of customer problems."

"The SIG should exert pressure on Digital to upgrade the Software Maintenance Services to a reasonable level. We feel that the services offered are not as professinal as they should be. Too much time is being taken by our people to solve Digital's problems."

"SIG helps by being clearing house of gripes, workarounds, etc. - but still have long turn-around through sending in information and seeing it in print. Need online SPR/PATCH database."

"From the stories one hears, the software support is in a bit of a mess, coupled with their policy of selective publishing of patches. I have the feeling that you are hitting your head against a brick wall. If they improve the service it will cost more and some users (especially those on tight budgets) will drop out...The best you can do is push and shove them to improve their service, but I fail to see why the RSX SIG should become managers of DEC's software suppport. The other approach is for some brave company to collect the information and sue DEC for neglegance and really shake them up."

4.0 CONCLUSION

There were many more comments in the survey. I have tried to publish a representative sample and only lack of time prevents me from listing them all. There seem to be three major points to remember about Software Maintenance Services:

1. SMS works correctly much more than it fails. Many people are happy with some or all of the services and feel they are getting their money's worth.

2. SMS is a very large, complex system. There are no quick, free fixes that can be made. Anything that Digital could do to make the system better would cost you more. Anything the SIG could do would probably be about the same as punching a large bowl of jello.

3. Digital does need to remember how serious a software error is. If my RSX system does not work, either because of hardware or software problem, it is useless. SMS should strive for timeliness and completeness.

Fall '82 DECUS Symposium

AIRPORT TRANSPORTATION

Luxe Livery Service, Inc. has offered to provide our attendees Airport transportation to and from the Los Angeles Airport at a discounted fee. The total cost for our attendees will be fifteen dollars ($15.00) each way excluding driver gratuity.

You can arrange to be met at the airport outside your airline by a uniformed driver who knows your name and where you are going. The driver will load your baggage and take you directly to your hotel.

FROM LOS ANGELES AIRPORT

After claiming your baggage please proceed to the center island, across from the baggage claim area. The van service is only permitted to park to load passengers. If you do not see Luxe Livery's green van in your area, watch for them in traffic - they may be circling the airport. To be recognized by the driver detach and wave the green portion of your transpass as he approaches. In the event you do not make contact with your driver within 10 minutes, please phone (800) 422-4267 or (714) 558-1413 and ask to speak with the dispatcher.

TO LOS ANGELES AIRPORT

Call (714) 558-1413 after 7:00 p.m. the evening before service to confirm your reservation and pick-up time. Meet the van at the hotel's main entrance. Be prompt! The drivers are not permitted to wait more than 10 minutes. Remember, Airport check-in lines are usually quite long.

SPECIAL NOTES:

1. Acquire a transpass from the DECUS office. A special mailing will be done for all preregistered attendees and DECUS Leadership. Anyone else wishing to recieve the transpass must call the DECUS Office at (617) 467-4875.

2. Each attendee is required to have their own transpass. Therefore, when requesting transpasses from the DECUS Office please state the correct number you need.

3. Phone Luxe Livery Service and make an advanced reservation. The phone number for outside California is (800) 854-8171 and within California (800) 422-4267.

4. The transpass must be presented to the driver at the time of transfer to be eligible for the discounted rate.

5. The transpass has no value in itself, cannot be bought or sold, and must be used in conjunction with CASH ONLY at the time of transfer.

6. Please pass this information on to your SIG/LUG members so they can also benefit from this offering.

DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MR02-3/E55
MARLBORO, MASSACHUSETTS 01752

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 129
NORTHBORO, MA
01532

ASSOCIATE [redacted]

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: _____

Mail to: DECUS - ATT: Membership
One Iron Way, MR02-3/E55
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here. Include name of installation, company, university, etc.