THE MULTI-TASKER

Volume 14, Number 1 January 1981

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

Contributions should be sent to: Editor, The Multi-Tasker, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752

European members should send contributions to: Colin A. Mercer, Tennant Post, High Street, FAREHAM, PO16 7BQ,
Hants, England

Members in Australia or New Zealand should send contributions to: Clive Edington, CSIRO, Computing Research
314 Albert St., East Melbourne, VIC 3002, Australia

Letters and articles for publication are requested from members of the SIG. They may include helpful hints, inquiries to other users, reports on SIG business, summaries of SPR's submitted to Digital or other information for the members of RSX-11/IAS SIG.

All contributions should be "camera-ready copy" e.g. sharp black type in a 160x240 mm area (8 1/2" x 11" paper with 1" margins) and should not include xerox copies. If you use RUNOFF to prepare your contribution the following parameters have been found to be satisfactory:

.PAPER SIZE 60,80 .LEFT MARGIN 8 .RIGHT MARGIN 72 .SPACING 1

These parameters assume output on a lineprinter with a pitch of 10 char/inch. Adjust the parameters to maintain the same margins if another pitch is used.

TABLE OF CONTENTS

Columns

From the Editor. 2
Speak Out. 3
Software Performance Reports 4
Hints and Things 7
Help Yourself. 9

Articles

DIGITAL Standard Runoff. 11
IND Patch. 13
System Tuning, Performance Optimization. 15
Chicago RSX-11M Q and A Update 26
European IAS Q and A Transcript. 30
CDA Questionnaire. 33

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

FROM THE EDITOR

The highlight of this issue is an excellent report from Jim Downward on his Workshop on System Tuning and Optimization. This paper covers the techniques Jim uses to make the net throughput of his RSX-11M system 2 to 4 times the standard system. The procedures can be applied to almost any system.

Other articles include an update from RSX-11M development team on the Question and Answer transcript published in the November, 1980 issue and a transcipt of the IAS Question and Answer session from the 1980 European Symposium. There is also a description of the new DEC Standard Runoff utility and a useful patch to the Indirect Command File processor from Marvin Brown.

The issue also starts what I hope will be regular columns. In addition to the traditional Software Performance Reports section, there are columns named "Help Yourself", "Hints and Things", and "Speak Out". "Help Yourself" is a question and answer section. In the past, the Multi-Tasker carried letters from users looking for information. These letters have been combined into this column. In addition, I will publish any answers received so everyone will benefit. The section "Hints and Things" will carry any short hints to make life easier, rummors to be passed along, and tricks to play on your system. Finally, "Speak Out" is a forum for all RSX/IAS SIG members to publish their opinions on the issues confronting us.

Please feel free to submit your input for these sections. In addition, the Multi-Tasker will continue to need and publish other articles and useful programs.

One long term goal for the Multi-Tasker is to improve the print quality and get more information in each issue. As an experiment in this issue, I have used DSR on my DECsystem-10 and a Diablo terminal for some of the articles. Also, we are trying the use of DECUS's headlining machine for the first time. I would be interested in any comments on this format. Some of the benefits I see are less white space and uniform print quality. Also, because some of the articles are in machine-readable form, they can be submitted to the SIG tapes so sites can produce their own back-issues.

Ralph Stamerjohn

P.S. A last minute addition to this issue is the CDA questionnaire at the end. A fellow worker at Monsanto, Ken Johnson, and I are planning a pre-symposium seminar at Miami on Crash Dump Analysis. This questionnaire will help Ken and me plan our presentations to be as effective as possible. Please take a few minutes to answer the questions.

SPEAK OUT

"Speak Out" is a monthly column for readers to express their opinions or to comment on a previous column. The articles published in this column are an individual's viewpoint and do not necessarily reflect the opinion of DECUS or the RSX/IAS SIG. Readers are welcome to submit articles on any subject concerning the RSX/IAS world. Submissions may be editted by the Multi-Tasker staff for space considerations and clarity.

RSX-11 AUTOPATCH SERVICE

Ed Chadwick
Western Electric Co.
Columbus, Ohio

The release interval on the RSX-11M Autopatch kit (QJ-729) is not acceptable. We purchased this package on the basis of quarterly releases (which is still the frequency that local salesman use), however, actual delivery has been about half that rate.

The most recent release is a prime example. Dates on the distribution indicate completion by the developers in late February and creation of the SDC master in March. But delivery to customers was not until August. SDC workload notwithstanding, an Autopatch kit has little value if it is months late in getting to the customer.

There is no reason that the Autopatch kit could not be released monthly, especially in view of the fact that local software specialists receive machine readable patches on that basis. It can be conceeded that a monthly release would likely be an overkill for the average customer; but, in view of monthly availability to software specialist, there is no excuse for not meeting the advertised quarterly rate. Further, there is no excuse for the release being several months out of date.

DIRECTIVE CHAINING

H.L. Coleman
Phillip Morris, U.S.A.
Richmond, Virginia

It is frequently desirable to execute two or more system directives in a sequence without having a context switch occur between those directives. This situation occurs most frequently when two tasks must synchronize their activity. For example, task A wishes to unstop task B and then stop itself until task B has finished its activity. If task B is of a higher priority than task A, a context switch occurs immediately after the unstop from task A is issued. There is no way for task A to then stop itself unless task B performs some type of wait activity.

Getting two or more tasks synchronized such that there are no timing windows is a very common situation in real-time systems. While RSX/IAS has some directives to aid this process, for example receive or exit, allowing directives to be chained together would completely eliminate the problem and give the application programmer enourmous flexability in designing his system.

There are many possible methods for implementing this feature. One would be a new form of the $S directive macros. A new argument would be added to each macro to specify that this directive is chained to the next or the directive is the last on the chain. Each macro would push its arguments on the stack and the executive would execute each in sequence and pop the user's stack. A second method would be a new directive (CHAIN$) which would take as arguments a list of directives to be executed.

In any case, the list of directives would be executed consecutively. A context switch would only occur if a directive on the chain is executed which places the task in a blocked state, Also some convention would need to be implemented to handle the directive status word (DSW) and what occurs if one of the directives in the chain is unsuccessful.

SOFTWARE PERFORMANCE REPORTS

This section contains SPR's submitted to the Multi-Tasker by users. SPR's should always be sent to DIGITAL. However, if you feel that a report should be published in the Multi-Tasker, you may send a duplicate copy to the editor at the addresses listed on the cover. Publication of an SPR in the Multi-Tasker does not imply endorsement by the SIG. Implementation of suggested fixes must be at the reader's own risk. The SPR's published in this column may be abstracts of the original submission and have not been checked for accuracy.

The following SPR on the RSX-11M V3.2 memory allocation algorithm was submitted by Mark Johnson (Research Council of Alberta, 11315 87th Avenue, Edmonton, Alberta, T6G 2C2).

> The RSX-11M V3.2 memory allocation algorithm is insufficient for certain circumstances. When the executive attempts to allocate space in a system partition, it first searches for a hole large enough for the new task. If this fails, the partition is rescanned to see if there is a large enough checkpointable task (and hole). If this succeeds, a checkpoint request for the task is made and the algorithm stops.
>
> The problem occurs when the found task has outstanding I/O which takes a long time to complete, for example unbuffered terminal input. The partition will not be re-allocated until either the input completes or another task exits. If the partition happens to be GEN, the system is locked up. This may not be a hard bug but it is an undesirable feature.

The following SPR on the RSX-11M V3.2 full-duplex terminal driver was submitted by Dr. K.D. Jahn (Ford-Werke AG, Abt. MC/PS-5, Postfach 60 40 02, Cologne, West Germany).

> The RSX-11M V3.2 full-duplex terminal driver does not use the initial timeout count stored in the terminal status control block (SCB).
>
> This leads to a problem when using certain terminals, particularly Tektronix scopes, which are modified to have the transmission line clock supplied by the terminal and not by the computer interface. When the screen is full, the terminal halts the line clock and renables it when the screen is erased and ready for more output from the computer. However, the terminal driver reports "DEVICE NOT READY" error because the timeout for character transmission has expired.
>
> This is a normal condition that was resolved in RSX-11M V3.1 by modifing the terminal initial timeout value (S.ITM) in the SCB to either a large value (255 seconds) or zero (no timeout timing). This patch does not work anymore for the RSX-11M V3.2 full-duplex terminal driver. The timeout occurs no matter how S.ITM is set.

NOTE

See HELP YOURSELF column for a possible answer to this problem.

The following SPR on RSX-11M V3.2 PIP was submitted by David Kristol (Massachusetts Computer Associates, 26 Princess Street, Wakefield, MA, 01880).

> PIP always appears to detach from the user's terminal, even if it did not attach to the terminal for output.
>
> This causes a problem that occurs very frequently on our system. A user starts a PIP operation and before it completes, begins another program like TECO that attaches the terminal. Since the terminal is attached, the PIP detach request hangs PIP until the terminal is detach by the second program. System performance degrades because of the frozen, non-swappable, non-shuffable copy of PIP. This problem did not occur in RSX-11M V3.1.

The following SPR on the RSX-11M V3.2 CO: driver was submitted by Rodger Miles (Telemed Cardio-Pulmonary Systems, 2345 Hoffman Estates, Illinois, 60195).

> The output from tasks writing to CO: can be lost.
>
> The CO: driver has its own internal buffer in which it places messages to be output by the COT task. During times of high activity, this buffer will fill up if COT does not empty if fast enough. If this happens, the driver will issue an undocumented error code IE.NOD to all tasks issuing a write request to CO: until COT issues a read request to empty the internal buffer.

The following SPR on the RSX-11M V3.2 disk drivers was submitted by Lance Szyhowski (McDonnell Douglas Electronics Co., Dept. 1492, 2600 N. Third Street, St. Charles, MO, 63301).

> A disk error can occur if an I/O request is issued to a RL01 disk immediately following the completion of power fail recovery. This problem is common to most other disk drives.
>
> The problem only occurs if the controller was idle prior to a power failure and a new I/O request is queued to the driver before the disk drive has spun up to ready after power recovery. The error is the driver logic only allows one 15-second interval for the disk to become ready instead of the normal four 15-second intervals that take place if the controller had been busy when power failure occured. The following series of events occur in the RL01 driver when power recovery is followed by a I/O request.
>
> 1. The driver powerfail code is entered and US.SPU is set in the UCB to indicate powerfail has occured on the unit. The busy status in the SCB (S.STS) is not touched if the controller was idle.
> 2. An I/O request is issued to the driver. When dequeued by $GTPKT, the controller is set busy (S.STS = 1). The driver checks to see if the drive is ready and when it is not, exits and sets one 15-second time-out interval.
> 3. When the time-out occurs 15 seconds later, control is passed to label DLOUT. It sees that S.STS is a one and clears US.SPU. It then attempts to retry the operation.
> 4. On retry, the disk drive is still not ready. The driver checks US.SPU and since it is no longer set, declares an error.
>
> As seen from the above sequence of events, the error occurs because S.STS is not properly set to four when the packet is dequeued by the driver. The following patch will allow DLDRV to check if the unit is spinning up and properly set S.STS when a packet is receieved. It should be noted that this same problem exists in several other disk drivers I have looked at (DKDRV, DBDRV, etc.).

```
-229,229,/;LRS/
1$:                                     ;REF LABEL

          .IF DF   P$$RFL

          BITB     #US.SPU,U.STS(R5)    ;TEST IF UNIT SPINNING UP
          BEQ      2$                   ; IF EQ - ALREADY UP
          MOVB     #4,S.STS(R4)         ;ADJUST BUSY COUNT
2$:                                     ;REF LABEL

          .ENDC

          MOV      R5,CNTBL(R3)         ;SAVE ADDRESS OF UCB
/
```

The following SPR on the RSX-11M V3.2 UNL command was submitted by Terry Palmberg (Telemed Cardio-Pulmonary Systems, 2345 Hoffman Estates, Illinois, 60195).

A user-written driver that does not have a unit 0 cannot be unloaded. The following error message is displayed, however, there is no documented restriction that a driver have a unit 0.

```
UNL -- DEVICE dd0: NOT IN SYSTEM
```

An examination of the UNL code shows that it assigns a logical unit to dd0: of the device being unloaded. If unit 0 does not exist, the assign returns an error and UNL fails.

The following SPR on SORT-11 V2.0 was submitted by Leon Keller (M&S Computing, P.O. Box 5183, Huntsville, Alabama, 35805).

SORT fails whenever the last byte of the last record of a file ends exactly on a block boundary and the file has more blocks allocated than used. The following error message appears:

```
SRT--P:?-INPUT ERROR ON dev:filename.ext
```

A work-around solution is to copy the file with PIP in order to get rid of the unused blocks.

The following SPR on RSX-11M PLUS V1.0 was submitted by Leon Keller (M&S Computing, P.O. Box 5183, Huntsville, Alabama, 35805).

The FORTRAN interface routines VSDA and VRDA (variable-length send/recieve) has three separate problems:

1. If the first word of the buffer being sent contains a minus one (177777), the received message is garbage.

2. The routine VRCD does not return DSW status if the IDSW variable orginally contains a minus one.

3. The documentation for VRCD indicates that IS.SUC is returned upon successful completion. In fact, VRCD sets the DSW to the positive number of words received.

HINTS AND THINGS

"Hints and Things" is a monthly potpouri of helpful tidbits and rumors. Readers are encouraged to submit items to this column. Any input about any way to make life easier on RSX/IAS is needed. Please beware that items in this column have not been checked for accuracy.

The following items were collected and submitted by Colin Mercer from various sources in Europe.

MAGTAPE ACP

It appears that for multiple controller magtape configurations, messages appear with the wrong unit number. A new set of objects are to be put onto an autopatch kit to eliminate the problem. The error is also related to magtape error messages appearing on random terminals.

EDITOR'S NOTE

See the RSX-11M December, 1980 Software Dispatch for an article on this problem.

EDT CRASHES WITH %ALL

If the %ALL range is used after an INCLUDE command, then EDT crashes. This is an overlay problem and is fixed by editting EDTBLD.ODL and rebuilding EDT. Change the OV1 line in the overlay description file as follows:

```
    OV1:    .FCTR    EDT-EDTLIB.OLB/LB:BLILIB:WFQIO:GETREC-OV1A
to
    OV1:    .FCTR    EDT-EDTLIB.OLB/LB:BLILIB:WFQIO:GETREC-GETFL1-OV1A
```

Also change the MSC line as follows:

```
    MSC:    .FCTR    MISC-*(CM3Y,M1,M2,M3)
to
    MSC:    .FCTR    MISC-*(CM3Y,M1,M2)
```

And eliminate the M3: line as it is not needed anymore.

VARIOUS LPA PROBLEMS

The following problems and rumors have been heard about the LPA device.

- The LPA FORTRAN support software for servicing AST's does not save the floating point registers.

- It is believed a new microcode update for the LPA is now available which cures some of the 'interesting' features of the LPA.

- Rumour has it the shuffler will on occasion move a task doing a long LPA acquire function.

SORT V2 AND COBOL V4

SORT treats large binary keys (more than one 16-bit word) in the reverse order to COBOL. A patch is thought to exists.

BASIC-PLUS-2

The BASIC-PLUS-2 installation kit tries to run [1,10]RESSTB.TSK from the distribution disk (which is or should be write protected). Change BP2BLD.CMD to just issue RUN RESSTB as the task has already been copied to the target system.

HELP YOURSELF

"Help Yourself" is a place for you to get your tough questions answered. Each month, questions from readers will be published. If you have a question, send a letter to the Multi-Tasker at one of the addresses listed on the cover.

We would also like to publish the answers to questions. If you can help someone, send a letter to the Multi-Tasker or call Ralph Stamerjohn at (314) 694-4252. Your answer will be sent directly to the person in need and published in the next edition of the Multi-Tasker.

ANSWERS TO PREVIOUS QUESTIONS

FULL-DUPLEX TERMINAL DRIVER OUTPUT TIMEOUT

Dr. K.D. Jahn submitted an SPR published in this issue about the RSX-11M V3.2 full-duplex terminal driver not using S.ITM for output timeout. I looked at the driver and think I have an answer. My examination shows that output timeout is done using offset U.TOTI in the secondary terminal block. This location is initialized at 2 points in the driver to 5 seconds. The specific instructions are in modules TTODN.MAC and TTRW.MAC.

```
MOVB    #5,U.TOTI(R5)           ;TTODN, line 239

MOVB    #5,U.TOTI-U.TOP(R5)     ;TTRW,  line 288
```

I believe that by changing these instructions to a larger value the timeout period can be lengthened or by setting the byte to zero, output timeout can be disabled. I have not tried this so if someone verifies if it works, please let me know.

Ralph Stamerjohn

SOLVING THE SOFTWARE DISPATCH LABYRINTH

At the San Diego Symposium, many sites complained they were not getting their Software Dispatches or were getting too many copies. The primary problem appears to be errors in DIGITAL's new database for software subscription services. Angela Cossette, ASG Manager - Software Services, guaranteed to resolve any problems reported to her.

If you are not getting your Software Dispatches or are getting someone else's copy, please send a letter to Angela. Please include any supporting documentation such as the Dispatch mailing labels and copies of the orginal purchase orders. Angela will get back to you when the problem is resolved. Her address is Angela Cossetts, P.O. Box F, Maynard, Massachusetts, 01754.

THE MONTH'S QUESTIONS

IAS TERMINAL LOCKOUT

During the past several months, a condition has been occuring on our IAS system where all the terminals are locked up. When such a lockout occurs, any attempt to enter input results in a bell, like the terminal's input buffer was already full. The only remedy is to reboot the system.

This lockout condition has occured at least once every three weeks, with as many as seven on one day. No clear pattern has been determined. The lockouts have occured when the system was quiescent as well as with a heavy load. In terms of hardware, it all checks out and no changes had been made at the time the lockouts began. Finally, static was suggested so all devices were grounded and static mats installed, all to no avail. Can anyone help us solve this problem?

Nancy Barry, NYCOTBC, 1501 Broadway, New York, New York, 10036. Phone (212) 221-5026.

8085 CROSS COMPILER

I am looking for a "C", Pascal, or equivalent structured language compiler which will run on RSX-11M and generate machine code for an 8085 micro processor. If anyone knows of such software, I would be please to hear from them.

Bob Martin, Teradyne Central, 3368 Commercial Avenue, Northbrook, Illinois, 60062. Phone (312) 291-4300.

SORT-11 AND FORTRAN

We use RSX-11M V3.2 with Fortran IV V2.2 and SORT-11. We would like to use SORTI and SORTA on variable length files and then use subroutines from a Fortran program to sequentially read the sorted index file and the read the orginal data. Could anyone supply me information on how this can best be accomplished?

William C.R. Webb, Business Data Services, 180 Winard Avenue, Sellersvile, Pennsylvania, 18960. Phone (215) 257-1161.

WHERE IS STANFORD SYSTEMS?

I am having trouble locating a vendor who supplied a piece of software to my employer. The software is an emulator package called UT2 written by Stanford Systems Corporation, 525 University Drive, Palo Alto, California. This package emulates a CDC 200 UT terminal from our PDP-11/34. Stanford Systems has disappeared and as far as I can tell, no one has assumed support for this software. I am interested in hearing from anyone who is familiar with this program, knows of what became of Stanford Systems, or has knowledge of an alternate vendor.

Richard L. Fischer, PORTEC, 300 Windsor Drive, Oak Brook, Illinois, 60521.
Phone (312) 325-6300.

DIGITAL STANDARD RUNOFF

This is a re-creation of the DIGITAL Standard RUNOFF (DSR) presentation made at the Fall 1980 DECUS Symposium in San Diego. The article was submitted by Al Lehotsky of DIGITAL's Software Tools group.

DSR Development History

- o Started in Fall, 1977 as an exercise in transportable BLISS.
- o More than 30 versions of RUNOFF studied:
  - Several MACRO-10 versions
  - Dozens of MACRO-11 versions running on RSTS/E and RSX
  - DECUS version for PDP-11's
  - Versions in BASIC
  - Independent MACRO-11 version running on RT-11
  - OS-8 version for PDP-8's
  - Other text processors (NRUNOF, etc.)

DSR Development Information

- o Based mainly on old MACRO-10 RUNOFF.
- o Merged useful features from many separate versions.
- o Written in transportable BLISS.
- o Is used daily within DIGITAL.

Compatibility

- o High priority issue. We wanted it to be easy to transport RUNOFF text files among DIGITAL operating systems. It was also important that existing RNO files be usable with DSR.
  - Command syntax should be upwards compatible
  - Resulting output should be unchanged

- o Tested against over a hundred existing RNO files on the DECsystem-10.
- o Fully compatible with old MACRO-10 RUNOFF, except for:
  - No .TYPESET commands
  - No .SELECT capability for generating program logic manuals from program source files

- o Accepts old PDP-11 RNO syntax with minor differences in how semicolon is handled.

Use of RUNOFF in DIGITAL

- o Technical Documents:
  - Software Manuals
  - Hardware Manuals
  - Writer's Drafts
  - Final documents produced on DIABLO's

- o Specifications
- o Project Plans
- o Memos

DSR New Features

- o Automatic table of contents generation (TOC utility).
- o Automatic two-column index (TCX utility).
- o .REQUIRE files include text during "compilation".
- o Greater flexibility in lists and numbering:
  - Number, bullet, upper-case, lower-case, or roman numerals
  - Nesting to many levels

- o More page layouts (.LAYOUT command) to control placement of title, subtitle, and page numbering.
- o More header flexibility (.STYLE HEADER command).
- o Conditional formatting:
  - /VARIANT:varstring command-line switch
  - .IF varstring
  - .ELSE varstring
  - .ENDIF varstring

- o New manual, DEC Standard Runoff (DSR) User's Guide, Order No. AA-J268A-TK.

Indexing Features

- o Comprehensive indexing features.
- o Includes all known indexing capabilities from old RUNOFF's:
  - Index flag ">"
  - .INDEX command
  - Subindexing to many levels
  - .ENTRY command supports "See Also" indexing
  - Control over indexing case translation rules
  - Bolded and underlined index entries

- o Can flag index "HITS" in the .MEM file with /DEBUG:INDEX switch.
- o Supports different styles of page numbers, including chapter-style (1-2) and Roman numerals (IV).
- o Single-column index with "......" available for old style indexes.
- o Optionally creates a separate index file (.BIX file):
  - NDXDMP utility dumps index hits in input order
  - TCX utility creates a "book-style" index
  - Can build large indices on PDP-11 systems by concatenating .BIX files.

Availability

- o TOPS-10
  - Supports KI-10, KL-10 processors
  - Unsupported, released with TOPS-10 V7.01
- o TOPS-20
  - Uses TOPS-10 command line
  - Unsupported, released with TOPS-20 V4
- o VAX/VMS
  - "Real" VMS command
  - Unsupported, released with VMS V2.0
- o RSX-11M / RSX-11M-Plus
  - Some limitations, .DO INDEX and .PRINT INDEX not supported
  - Only mapped systems with EIS supported
  - Now being field-tested internally
- o Other PDP-11 Operating Systems
  - RSTE/E planned
  - IAS planned
  - RT-11 is not planned

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

P. O. Box 3000 • Boulder, Colorado 80307

In the October 1980 issue of the Multi-Tasker, Colin Brewer showed a solution to a SIG Menu request by modifying the indirect command file processor (AT.) to take a command file from a default UIC in the same manner as the RUN directive. We also have found a solution to this problem.

We have several tasks which search the current UIC then a secondary default UIC. They include TECO and a modified RUN command which passes a command line. To compliment these programs, I have modified the indirect command file processor to alternately search for a secondary default. Attached is the SLP listing for the modifications.

Another feature which has proven to be very useful is a FORTRAN interface to the command line processor (GCML$) and the table-driven parser (TPARS). This interface is provided thru the package named COMMAND which was submitted on the FALL 1980 RSX SIG tape. With COMMAND, a complex command line syntax can be defined and data from command lines can be easily accessed from a FORTRAN program.

Marvin Brown
Systems Programmer
Research Systems Facility
Phone (303) 497-0681
        FTS 322-7681

```
[12,10]INDFIL.MAC;2=[12,10]INDFIL.MAC;1
-2,2
        .IDENT  /04.06+/
-38
;       MLB001  19-NOV-80       ADD SECONDARY INDIRECT FILE DEFAULTS
;
%
-129,,/;MLB001/
;
; SECONDARY DEFAULT NAME BLOCK
;
SFNB::  NMBLK$   ,CMD,,LB,0
SFNBU:: .WORD    SFNBC,SFNBI
SFNBI:  .ASCII   /[4,3]/
SFNBC   =        .-SFNBI
        .EVEN
/
[12,10]MGCML.MAC;2=[12,10]MGCML.MAC;1
-2,2
        .IDENT  /10.12+/
-58
;       MLB001  19-NOV-80 ADD SECONDARY COMMAND FILE DEFAULTS
;
%
-540,,/;MLB001/
        BCS      5$                     ;++002 SKIP ON ERROR
        OFNB$R                          ;++003 OPEN FILE FOR READ (VIA FNB)
        BCC      10$                    ; GOOD, SO DONE...
5$:     MOV      R2,-(SP)               ; SAVE DESCRIPTOR
        MOV       SFNB,R1               ; GET SECONDARY DEFAULT NAME BLOCK
        MOV       SFNBU,R2              ; GET UIC STRING
        CALL     .GTDIR                 ; PUT IT IN
        MOV      R0,R1
        ADD       F.FNB,R1              ; GET FILE NAME BLOCK
        MOV      (SP)+,R2               ; RESTORE DESCRIPTOR
        MOV       SFNB,R3               ; DEFAULT NAME BLOCK ADDRESS
        CALL     .PARSE                 ;++002 PARSE THE FILE NAME
-544,,/;MLB001/
;       SECONDARY DEFAULTS

        .ATDUI   = 003004
        .ATDLB   = "LB

/
[1,24]INDBLD.ODL;2/-AU=[1,24]INDBLD.ODL;1
-9
;
;       MODIFIED BY MARVIN BROWN (NCAR) ON 19-NOV-80 TO ADD GETDIR
;
-40,40
PRS:    .FCTR   LB:[1,1]SYSLIB/LB:PARSE:PPNASC:UDIREC:GETDIR
/
```

Workshop on System Tuning, Performance Measurement
and
Performance Optimization of an RSX11M System
November 11, 1980

James G. Downward
KMS Fusion, Inc.
P.O. Box 1567
Ann Arbor, MI 48106

ABSTRACT

Anyone who has done an RSX11M SYSGEN invariably wonders what real effect the various "performance" SYSGEN options have. Once one has a running system, soon one starts wondering if there are any other ways to improve the system's overall performance or functionality. This session will attempt to address these issues. Topics to be discussed by the panelists will include:

- How to define performance. How to measure it. Performance measurement tools.
- SYSGEN options. Setting executive timesharing intervals. QIO optimization. Choice of F11ACP. Useful tricks.
- POOL
  - a) Minimizing POOL fragmentation
  - b) Decreasing POOL requirements.
- Use of Resident Libraries.
- Optimizing disk usage and access times. Measuring system disk loading.
- Improving program execution times.
- Optimizing small/big system performance.
- Optimizing a system for:
  - a) program development
  - b) real time work.

Time will be provided for questions and discussion.

INTRODUCTION

This is going to be a workshop on System Tuning and Performance Optimization for RSX11M systems. While most of what will be discussed is particularly applicable to RSX11M V3.2 systems, some of it will be generally applicable to other systems. The first part of this session will be devoted to presentations by the speakers on a wide ranging series

of topics. This will take about 1.5 hours. Then in the last half hour we will welcome discussion, comments, suggestions and questions from the floor. We have a considerable amount of material to cover, so if questions can be held to a minimum, it would be useful.

I became interested in optimizing the performance on our RSX11M system over 2 years ago. We currently have an 11/45 with 11 terminals and a very large number of users, each of which covets the entire resources of our system. Two years ago we were critically tight on disk space and terminal time. This problem led me into disk block and terminal time accounting and finally into the area of resource accounting in general. After getting the disk space and terminal time problems under control, I found that the growth of the system usage was leaving POOL and memory critically tight. At this point I implemented Richard Kirkman's CCL, updating it for V3.1, and a whole raft of MCR, INDIRECT, and BATCH enhancements. For a while our memory and POOL problems vanished. Then the user's discovered how convenient the system was, and usage went up, again pushing us up against the proverbial wall. To make matters worse, V3.2 of RSX11M seemed to require substantially more core than before. This pushed us into reconfiguring the system to make heavy use of a FCS resident library, and to monitoring system performance and performance of various software subsystems. These measurements helped justify adding a CACHE memory and additional disk capacity. At present, performance measurement and capacity planning is our critical interest. We are trying to get a handle on the ultimate capacity limits of our machine. Users now want to increase the number of terminals from 11 to 16 or more. Serious thought is being given to connecting the PDP-11/45 to 3-4 satelite 11/23's via DECNET to handle program development for a new laboratory data acquisition system. At some point, a larger machine will be needed and hopefully performance measurements will give us enough information to plan our data acquisition system's growth.

By the end of this workshop, we hope to present material which will show you, the user, that very substantial performance improvements are possible. Depending on the amount of effort one wishes to invest, 10% to over 50% improvement in performance is not unreasonable. The goal of this workshop is not to tell you how to make these improvements, or to overly extol the virtues of any one method, rather, we want to tell you what can be done, give you an estimate of the degree of benefit involved, and give some feeling for the complexity of the task and any system consequences. All of the improvements I will discuss, we have implemented on our system. The net result is that throughput on our system is perhaps 2-4 times greater than on a standard 11/45 running a standard RSX11M operating system. The lower limit to throughput improvement is the result of adding a CACHE memory. The upper limit is the result of many software and system changes.

What do I mean by performance? Performance can either be considered system wide or on a per task basis. A development system will probably be concerned with overall optimization of the number of tasks processed through the system. A dedicated system will be primarily interested in optimizing performance of a set of tasks. Toward this end we will address a number of topics.

TYPES OF OPTIMIZATION

1. Overall System Throughput
2. Subsystem or Application System Throughput
3. Ease of use (system/subsystem)
4. Memory Usage
5. Pool Usage (previous session)
6. Disk Usage

(SEE SLIDE OF TOPICS AND SPEAKERS)

TUNING A SYSTEM

A) HARDWARE OPTIMIZATION

The easiest way to improve performance of an existing system is to install a CACHE memory if it doesn't have one already. I believe that one can now get CACHE memories for any DEC processor larger than a PDP 11/20 either through DEC or through a BRAND X company. For about $6K or so, your system can run 30-50% faster.

Swapping disks or disk emulators can also provide substantial benefit to a heavily loaded system. While we have not yet incorporated one on our system, other users have and claim that dramatic performance increases are possible depending on the application. A heavily overloaded system which does a lot of checkpointing clearly would benefit from such a device. Often just redistributing the disk usage on a multi-disk system can provide significant benefit. The trick is to find out where 'bottle necks' are occurring and to know which changes help the situation the most.

For instance, if one has multiple disks, it may be advantageous to move the checkpoint file to a disk other than LB:. When a system gets busy with many tasks loading from LB:, bringing in overlays, and searching through object module libraries during taskbuilds, checkpointing to LB: only leads to further disk head contention. By placing the checkpoint file onto another disk (even if it is a slower disk) disk access time and checkpoint speed may significantly increase. This in turn may lead to increasing system throughput under heavy load conditions.

Finally, users doing a lot of high speed terminal I/O can improve their system's performance quite significantly by using a DH11 (DMA) terminal interface and the V3.2 full duplex terminal driver. Using a DH-11, reduces the amount of terminal driver overhead required for processing an interrupt for each character output. The overhead for a single task doing continuous output at 9600 baud is quite considerable (See Art Perlo's letter to the MultiTasker in 1980).

B) SOFTWARE AND OPERATING SYSTEM OPTIMIZATION

Without resorting to using performance measurement tools or incorporating hardware enhancements, there are a number of software options the user has readily available which will improve system performance.

1) SHUFFLER and RMDEMO

The SHUFFLER and RMDEMO affect the performance of a heavily loaded system such as ours. Firstly, comparing two identical loaded systems except that one is running the new RMDEMO and the other is running the old RMDEMO, one finds that the new RMDEMO imposes at least a 25% greater overhead than the old version. Part of this is because it is larger and more swapping occurs. Secondly, the new RMDEMO issues 1 QIO for each line changed on the screen since the last plot. On systems selecting checkpointing both on input and output, RMDEMO can get into a mode in which it checkpoints between QIO's if the system is loaded. In some cases the system will swap itself to a standstill. For those who wish to use the old RMDEMO, it is on the SAN DIEGO tape from the last DECUS symposium. It is modified for V3.2, and has many of the features of the new RMDEMO. We plan to support it at least through V4.0.

A second, very useful trick for a heavily loaded system is to place SHF in its own partition and FIX it there. This costs almost no memory, since SYSPAR is much too large as it stands for a mapped system. It can almost hold SHF and MCR... side by side as it is now. Tests of system throughput for a system doing a lot of swapping show that throughput is increased by about 5% if SHF is fixed in its own partition.

If you have an extremely heavily loaded system, you find that a large fraction of the systems time is spent with SHF trying to make room and not being able to. It is a design 'feature' of the shuffler that there is no limit placed on the number of times/sec it will try and recover memory to run a task. There is a relatively simple way around this problem if one wishes to modify the EXEC slightly. Each Executive Swapping interval, load a counter with the maximum number of times one feels SHF may run during a swapping interval. Then each time the SHF is requested, decrement the counter. If the counter goes to zero, reject the request. One should also have the SHUFFLER get a flag to tell the EXEC it is active but waiting since it will only make matters worse if SHF gets requested part way through a run. We have done this on our system and gained 10-20% more throughput out of a thrashing system.

2) QIO OPTIMIZATION

If you have selected QIO optimization at SYSGEN you also have another easily modified tuning parameter. As a quick and dirty test of what one might expect, I dumped an RK05 to NL: using PIP with MAXPKT set to 15 and 0 respectively. I found that it took 2.6% less time to complete the transfer with MAXPKT set to 15. Next, I wrote a test program which wrote 10,000 records to NL:, installed it 6 times with different names, ran all six tasks at once, and timed how long it took for all six tasks to finish and exit. If MAXPKT was set to 15 rather than 0, the process took 6.6% less time to complete. This benefit would be less pronounced if one was transferring to a real device. But it is indicative that the system may be spending 5% or so less time servicing the QIO before it gets to the driver.

3) ROUND ROBIN SCHEDULER AND SWAPPING INTERVAL

Another tuning parameter which can be tweaked, is the Round Robin and Swapping interval. As part of the KMS System Accounting and Performance Measurement package, I included a task for displaying and modifying these intervals on-line. To test the effects of heavy swapping on throughput, I started up 6 simultaneous taskbuilds with versions of BIGTKB which were 23K in size. Only 3 of the TKB version could fit in core at one time, and the system was running with anywhere from 80-102K of tasks swapped out. Holding the Round Robin interval fixed at 5/100 second, the Swap time was varied from 20/100 to 1 sec. The results are displayed in the graph. The baseline is the time it would take for 6 sequential taskbuilds to take place. As you can see, as the swapping interval is lengthened, the time for the entire taskbuild approaches the baseline asymptotically. The 'dots' represent the timings taken with SHF in SYSPAR, and the "x's" represent timings with SHF fixed in SHFPAR. These timings were done for an RK07 disk. From them we see that a swapping interval of 40/100 of a second is about the minimum we should use. A slower disk (RK05) should use a somewhat longer swapping interval.

4) MEMORY USAGE

On our system, memory is very tight. In the face of an increasing number of users and terminals on our system, it was necessary to examine how tasks used memory, very critically. One of the first things that was noticed was that a significant number of DEC taskbuild command files for mapped systems build utilities and privileged tasks with default sizes which are far, far in excess of what is needed. A classic case in point is that the new despooler task was built with a task size of 8k words when it only needs 2-3k words. A second example is INS.TSK which is also built to be an 8k task. If it is installed via VMR with an INC=0, or taskbuilt to the correct size, its size is reduced by a little over 1k word. By going through the command files, and modifying the partition size prior to taskbuilding, you will create tasks which use less memory.

Some tasks, however, like BIGTKB use the extra space for symbol storage or buffering. For TKB what we did was to find the size required for a small taskbuild and start out TKB at that size. Since our system supports the extend task directive, TKB will grow in size if needed. Since having to do too many extend tasks slows down performance, use a CCL command to create a super large version of TKB for use during SYSGENS and RMS taskbuilds. This technique saves about 3k of memory for a typical use of TKB.

5) RESIDENT LIBRARIES

Since the taskbuilder is one of the biggest users of core, a substantial overall improvement in system throughput can be accomplished by making it run faster. This is one of the advantages of using resident libraries. Taskbuilding a given task using a resident library often takes only a small fraction of the time required for taskbuilding without a resident library. Those of us who have taskbuilt RMS tasks are well aware of how long task builds using the RMS ODL's can be. On a small system such as ours, the length of the taskbuild times insures that if other users are doing program development from other terminals, checkpointing is almost certain to take place. This problem is considerably alleviated by using an RMS resident library. However, a small system such as ours just can not support an extra 4-12k of resident libraries all the time.

There are two ways to circumvent this problem. First, one can use the SET/TOP command to shrink GEN and make room for a temporary (but otherwise very normal) resident library. The required software patches to MCR for doing this were published in the MULTI-TASKER a few months ago and are on the San Diego tape. This method is only useful during program development and is not transparent to user's.

A second and much better way of doing this is to use transient resident libraries. Brian Mcarthy (DEC) developed a method of loading resident libraries into PLAS regions. When a task is run, the resident library is automatically loaded into core and linked to the task(s). When the task exits the resident library PLAS region vanishes. The code to produce loadable resident libraries will be on this DECUS tape. We have been using it in production for 5 months and it has worked very well. There are two drawbacks. The first is that PLAS regions can not be shuffled. The second is that the method does not work on PIC libraries or on BP2 libraries linked with separate RMS libraries. However, a combined BP2/RMS 12k library is supplied.

6) BATCH

Another thing which can be done to improve overall throughput on a crowded system is to implement BATCH. An improved single stream BATCH and a multi-stream program development queue utility will be on this DECUS SIG tape. Numerous in house tests have shown that 4-8 users all trying to compile or taskbuild at the same time will swap the system almost to a standstill. Under conditions like these, it is prudent and to everyone's advantage to submit taskbuilds and compilations to a single or multiple stream BATCH queue. While working to optimize BATCH, we realized that the indirect file processor while a great tool, is very inefficiently used if it has to keep swapping in and out of core during the course of a program compilation and taskbuild. Since the majority of program development work uses simple MCR commands, we implemented a Procedure Interpreter (PIN) 4 times smaller than ...AT. which spawns off simple commands to MCR from a command file. Indirect is used on the fly to create the user specific PIN command files. Even if PIN gets swapped out, it is far more efficient to swap it in and out instead of ...AT.

7) CCL

I don't want to say too much about CCL. Once I get started it's hard to stop. It was origionally developed by Richard Kirkman, and modified by KMS and various RSX SIG members to adapt and enhance it for V3.2. CCL is not quite a panacea to all your problems, but it comes very close. It consists of two parts. A user extensible, file driven catchall task and modifications to INSTALL to allow passing MCR command lines to uninstalled tasks. The biggest benefit of CCL is to free up POOL. Most tasks, which were previously installed, now can be removed transparently to the user. Command lines are parsed and sent on from CCL to non-installed tasks. In addition, it can make your system unbelievably friendly. Simple user understandable commands can be used to aid the user through the complexity of our RSX wonderland. For example, we have a BUILD command. The user types BUILD FILNAME, and in a completely transparent way a FORTRAN task is correctly compiled and taskbuilt. Of any of the CCL's available on DEC systems (RSTS or VAX) this is perhaps the most powerful. I firmly believe that if you haven't got CCL on your system yet, you should run not walk, to your nearest San Diego Decus tape and get a copy to put on your system.

8) FCS RESIDENT LIBRARY

Building a system based on using an FCSRES is perhaps the hardest of the various performance/tuning options I have discussed. However, it holds the greatest potential for DRAMATICALLY improving your system performance. Among its benefits are:

BENEFITS OF USING FCSRES

1. Faster taskbuild times.
2. Smaller on disk task size.
3. Smaller in-core task size.
4. Tasks less overlayed, run faster.
5. LESS LOADING on the system disk.
6. Smaller tasks swap faster.
7. More tasks can fit in core.

DISADVANTAGES

1. More task to build for SYSGEN.
2. On line SYSGEN from one base level to another is hard.

On the 1979 San Diego Tape, I supplied command and ODL files which will allow building all DEC unprivileged utilities, and a few privileged tasks with FCSRES. In the table is a comparison of on-disk task sizes of DEC utilities built with and without FCSRES. For just these tasks there is savings in disk blocks of 23%.

COMPARISON OF TASK SIZES (WITH AND WITHOUT FCSRES)

	SIZE (Disk Blocks)	
TASK	No FCSRES	With FCSRES
BIGMAC	71	57
BIGTKB	161	145
CDA	159	114
CMP	50	29
CRF	36	24
DMP	57	41
EDI	60	41
EDT	108	88
FLX	129	106
FMT	65	57
IOX	99	79
LBR	72	52
PAT	44	25
PIP	67	51
SLP	48	30
VFY	57	39
VMR	144	124
ZAP	38	25
TOTAL	1465 blocks	1126 blocks

The amount of incore size reduction for DEC utilities built with FCSRES is not as large as one might hope for. The savings runs from about .5 to 2k words with the average around 1k word. The reason for this is that the utilities are very heavily overlayed. Also, the get command line code is not in the resident library (size??, PIC??) and is included in most of the utilities. Even so, consider the effect that a 1k word average size reduction has on a system with say 64k words of free core in GEN. Assuming an average task size of 8k words, 8 tasks can fit in core before swapping starts. If you lop off 1k words from each task, this frees up an additional 8 k words for an additional task which can be in core before swapping starts. An additional benefit is that initial task loads are significantly faster by an average of 13% for the typical utility. For the same reason checkpointing takes significantly less time.

The real benefit of building the tasks with a FCSRES is in the dramatic increase in speed of the standard DEC utilities. These utilities are normally very heavily overlayed. Building them with FCSRES removes almost all the FCS overlay in the task. Consider the following speed improvements.

SPEED IMPROVEMENT

PIP	NL:=DM:[*,*]*.*/FU	2.4 times faster
	NL:=[1,2]*.HLP	1.2 times faster
BIGMAC	NL:=HELLO,MAC	1.1 times faster
...AT.	For a file with a loop including: (.INC, .IF, .TESTFILE, .GOTO)	1.7 times faster

A very significant benefit is that the overall responsiveness of the system improves. Since the tasks have far fewer overlays, the system disk stays far less busy and task requests to read and write files are not interrupted by reading in overlays from the task image. At present, I do not have a reliable indicator of how much this affects performance, but on a busy RK05 system disk with 4-5 users at work with the utilities, the busy light flickers only intermittently if the tasks are built with FCSRES. If the tasks are not built with FCSRES, the busy light never goes off until the users exit.

Using an FCS resident library does have a few disadvantages. Firstly, one must rebuild all utilities for the initial FCSGEN. Normally, this need only be done once for each release cycle. Due to the number of tasks involved (both DEC utilities and our Fortran application programs) this task makes SYSGEN seem trivial. However, even it can be done on line in a day of continuous taskbuilding. However, if a bug in FCS appears which requires patching, FCSRES would have to be rebuilt along with all the tasks using it. This is not something to be lightly undertaken.

Because DEC has modified FCS at each release level, a FCS resident library must be rebuilt each time a new release of RSX11M is obtained. At present, building all tasks with a new version of FCS/FCSRES from an existing system using FCSRES is a bit of a pain. I managed to do it because I have multiple disks. I mount an alternate disk for the new SYSLIB and FCSRES and assign it as my local LB:. I then task build all utilities and application programs placing them on a UIC which will become my new LBUIC.

A user with only one large disk would have substantially more difficulty doing an on-line FCSGEN using the previous release of the operating system. However, such users have the same problem in general in doing an online SYSGEN from one base level to another.

PERFORMANCE MEASUREMENT

I have said a great deal so far about ways to improve system performance, but before you go trying to tweak your system to improve its performance, you must answer a very important question first, namely; "How do I know if or how much performance has changed?" Now our approach was evolutionary and piecemeal. I would make one change and then write a test program or procedure to test it out. For changes such as varying the value of MAXPKT or changing the Round Robin/Swapping times this approach is adequate. It does not however provide any standardized measure of how a system performs under load conditions.

The KMS accounting and performance measurement package provided information on performance but it lacked the ability to snapshot performance on a fine enough scale without gathering huge amounts of data. Also, there was no easy way to relate the data to the degree of system load. The selective task accounting feature easily enables one to measure the performance of a specific task as one attempts to optimize it. I felt that a similar system wide capability was needed for tuning system performance.

What was needed was a method of simulating the load on the system of 'N' users on 'N' terminals busily working away. In this way pool requirements, processor speed requirements, and disk access time requirements could all be estimated prior to committing to hardware and software. During the course of this test, I wanted to snapshot the statistics gathered by our accounting package, and to measure the activity on the system disk. I did not want to use physical terminals, since I wanted to observe the effect of having more terminals active than we currently have.

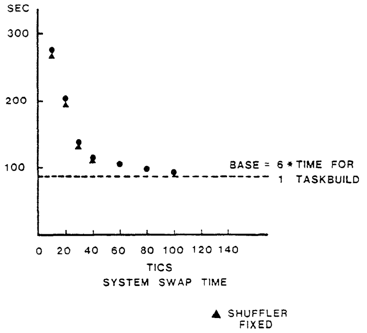
To do this, 'N' copies of INS were installed with names I00,I01, ...Inn. These versions of INS were used to start a procedure file running with a unique name for each procedure interpreter task (PINXnn). An indirect command file creates a unique procedure file for each PINXnn task. Each file contains commands of the form

```
Inn $F4P/TASK=F4PXnn/RUN=REM/PRM="FOO=FOO."
```

The indirect file processor monitors the status of the 'N' job streams and when all have exited, writes out all the statistics to a log file. For completeness, the command file also executes a program to snapshot our system performance statistics (CPU time, Shuffler count, Checkpoint count) before and after the load test. This data is passed to the command file which also logs these statistics. A separate task monitor's the busy time of LB: and this also is reported.

This performance measurement package has proved to be very versatile and useful. Using it, we were able to pin point a serious contention problem between TKB searching SYSLIB and tasks swapping in and out of the checkpoint file on LB0:. This package will be on the fall 1980 RSX SIG tape.

TIME TO COMPLETE
6 SIMULTANEOUS TASKBUILDS



ROUND ROBIN SCHEDULER AND SWAPPING INTERVAL

- TEST EFFECT ON THROUGHPUT OF SYSTEM WITH TWICE AS MANY TASKS TRYING TO EXECUTE AS THERE IS FREE CORE
- USE THE PROGRAM 'SCH' TO MODIFY ROUND ROBIN INTERVAL AND SWAPPING INTERVAL 'ON LINE'.
  (ON 1979 SAN DIEGO DECUS SIG TAPE)
- SYNTAX:

```
SCH/RBN:nn/SWP:mm
```

CONCLUSIONS

- THROUGHPUT INSENSITIVE TO ROUND ROBIN TIME FOR 'REASONABLE' VALUES (5 TICS OR SO).
- FROM GRAPH OF EXECUTION TIME VS. SWAP INTERVAL FIND THAT THE SWAP INTERVAL SHOULD BE 40 TICS OR MORE FOR AN RK07. THIS VALUE WILL VARY WITH DISK TYPE.

RSX-11M QUESTION AND ANSWER SESSION
CHICAGO, 1980 SPRING SYMPOSIUM

UPDATED ANSWERS FROM DIGITAL

The transcript of the RSX-11M Question and Answer session from the 1980 Spring Symposium held in Chicago was published in the November, 1980 issue of the Multi-Tasker. Since that time, DIGITAL has examined some of the questions more closely and updated some of their responses. The original questions and the new answers are published below. For further information refer to the original article.

8. When a program accessing a magtape is aborted, the tape does not stop moving until EOF is reached. Will this be fixed so the program aborts instantaneously and the tape left where it was?

   Space files operation cannot be stopped in hardware.

17. We are trying to use a loadable TT: driver with a loadable database for RSX-11M Plus. The system crashes when the first character is typed. Are there any inherent problems?

   An examination of the TT: database showed no reason the driver can't be loadable. We suspect that a bug was introduced when the database was made loadable.

23. Since the RSX-11M V3.2 full-duplex terminal driver attempts DMA transfers of the DH11, what is the problem in supporting 19.2K baud CRT transfers?

   Throughout at 19200 baud is only nominally higher than at 9600 baud due to driver latency.

30. Does DIGITAL have any plans to implement autobaud detection for the terminal driver in RSX-11M V4.0 or earlier?

   Autobaud is being considered for RSX-11M V4.0 and RSX-11M-Plus V2.0.

33. A BRU /DIR command generates junk in the listing. Will this be fixed?

   A patch has been published. See the August, 1980 Software Dispatch, article 5.1.17.7.

35. I have an extensive, online, real-time and control system that consistly does breakthough writes. It is 2-3 times slower under the RSX-11M V3.2 full-duplex terminal driver than using RSX-11M V3.1. Why?

   The difference in throughput was determined to be due to the task being checkpointable during the write breakthrough. This was a consequence of using the full-duplex terminal driver which can buffer output requests, as opposed to the half-duplex driver which cannot. There is a patch (unsupported) to disable checkpointing during output which was published in the September 1980, Software Dispatch, article 3.1.3.9.

38. What can one do about attachment of the line printer by LPP0 when the queue manager is running. The problem is that one cannot send anything directly to the printer as long as it is spooled. In answer, a patch by Jim Downward was mentioned that caused LPP0 to attach and detach the printer.

Do not use Jim Downward's patch (to stop the line printer despooler from keeing the printer attached) on RSX-11-Plus. In addition to being attached, the printer is set to spooled (DV.OSP is set in U.CW1 in UCB) and the printer appears to be redirected to the system disk. We are looking into a more general solution to the problem. A probable side effect of such a solution would be that detachable printers on RSX-11M-Plus systems would not allow transparent spooling.

39. What does SET /BUF do with a disk. I did a SET /BUF=DB:1024. and resulted in LOGIN.TXT being unreadble by HELLO and files being editted were lost.

In general, the buffer size of disks should not be changed from 512 if FCS is to be used to manage files on them. FCS uses the device buffer size to allocate it's internal buffers and as the transfer size in read and write QIO's. It is possible that there is not enought buffer space in the task to allocate the necessary number of double length buffers in the case mentioned, also possible that reads and writes do not correctly maintain the file attributes. Unfortunately, FCS must use the set device buffer size, to allow ANSI magtape to work. In short, don't change the disk buffer size.

40. Have past bugs in doing asynchronous FCS operations been fixed?

There is currently only one known restriction with FCS asynchronous file operations. Since FCS maintains only one DPB to use for all of the QIO directives it issues (the DPB is in $$FSR2), if an FCS operation at non-AST state is interruped by the completion of an asynchronous read or write, and the AST routine issues another FCS request, the AST call may re-use the DPB of an operation for which the DPB is currently being built at non-AST state, thus corrupting the DPB and causing the non-AST operation to fail. The work around for this problem is to do all FCS processing at non-AST state, and to use the completion routine to notify the non-AST state code of the completion.

44. BRU improperly handles large files (greater than 30,000 blocks).

Several patches have been published to correct problems with large files. See August 1980 Software Dispatch, article 5.1.17.8 and other articles on BRU.

49. On a heavily loaded system the following sequence of commands crashes the system: INS FOO, FIX FOO (not enough room), RUN FOO (task active), ABO FOO (task not active), REM FOO, and system crashes.

A patch has been published to correct the problem. See the September 1980 Software Dispatch, article 2.2.1.12.

52. The $LOCKL and $UNLKL subroutines in MCRDIS are no longer called by RSX-11M Plus. We have a need for a dispatcher of our own and use MCRDIS. Do I need to call $LOCKL in RSX-11M-Plus?

Most references to $LOCKL have already been removed from RSX-11M-Plus and RSX-11M will have them removed in V4.0. The references are being changed to calls to $SWSTK. Some references to $LOCKL may remain.

55. Is there any way to change your system disk on the fly on RSX-11M Plus?

There really is no reliable method of changing system disks on the fly.

59. My system is a PDP-11/34 running RSX-11M V3.2 with the full-duplex terminal driver and 4200 bytes of pool. When read with prompt is interrupted by a control-C, the terminal driver hangs 60% of the time by responding with a control-U to all input requests. An SPR was requested by DIGITAL.

No SPR was received, however an investigation showed that the application resulted in a severe lack of free pool. The full-duplex driver will echo control-U when not enough pool is available to queue a command line to MCR.

60. After outputting to LP: with DMP, subsequent attempts to dump to another device, like TI:, fail with message LUN LOCKED IN USE. Why?

This appears to have been a very old problem related to the fact that DMP occasionally forgot to detach an output device. It has (we believe) been corrected for some time.

62. Under RSX-11M V3.1, we are trying to use one of our two DZ11's, both known to the system, to capture data at 2400 baud. We have hopefully disabled the interrupts for that DZ11 and are doing direct physical I/O. We are only getting about half the data. This approach worked using a DL11-E.

During the discussion, it was pointed out that it is not a good idea to have a task poke any terminal controller which is known to the driver. Several other suggestions were made for solving the problem, including attaching the terminal and double buffering input requests.

63. The half-duplex terminal driver in "read special terminator mode" will not see escape sequences generated by the VT100 cursor keys. This makes it impossible to use the VT100 in a FORMS mode without resorting to single character QIO's.

The read with special terminators function will terminate a read when an escape character (the first character of the sequence) is received. Since the VT100 sends the rest of the escape sequence in rapid succession, it is possible to lose those characters when using the half-duplex terminal driver. Our suggestions is to attach the terminal for escape sequences (IO.ATT!TF.ESQ) and issue read virtual block functions. The read will not complete until the entire escape sequence has been received, thereby eliminating the window between the receipt of the first (escape) character and the rest fo the escape sequence.

70. Explain the shuffler algorithm. The answer stated the shuffler was being reevalated for the next release.

We requested, but have not received any input from Dan Steinberg, regarding shuffler evaluation.

73. In RSX-11M V3.1 or V3.2, does the terminal driver drop DTR for the DZ11 and DH11? We are planning to use smart Gandalf PACX.

Yes.

80. There is an apparent incompatability between the half-duplex and full-duplex terminal drivers. A read-all with echo QIO works properly in the half-duplex driver. The full-duplex driver does not echo correctly. For example, tab echoes as multiple spaces and formfeed echoes as multiple line-feeds.

SPR has been answered, but was not published.

87. Are there any plans to use the hardware I and D space capability for RSX-11M?

RSX-11M will NEVER use separate I/D space, this is the second largest difference (in terms of code size) between M and M-PLUS.

91. The full-duplex terminal driver supports HFILL and VFILL. This makes unbuffered hardcopy terminals work nicely with the exception of EDT sessions. Why does EDT have to maintain its own set of terminal characteristics?

This question was passed on to the EDT maintenance group, and will be considered for a future release of EDT.

95. Could DIGITAL print in the Software Dispatch the SLP checksums for all future and past patches.

As we said last fall in San Diego, we believe we can get SLP checksums wrong as often as we get PAT checksums wrong, and are therefore hesitant to use them. However, we have seen a marked improvement in our ability to publish patches with correct checksums, and will consider supplying SLP checksums.

96. Could you make IO.ATT and IO.DET return an error instead of queuing the request.

This may be implemented in the next releases of RSX-11M and RSX-11M-Plus.

97. FMT will not use the the manufacturer bad block for RM02 disks using CDC 9877FF packs.

An SPR was submitted and a patch was supplied.

101. When system runs out of pool, RMS files can become inaccessible.

A patch was published in the July, 1980 Software Dispatch.

102. Why is RMS-11 distributed with RSX-11M V3.2 instead of RMS-11K. RMS-11 is the same functionally as FCS.

We are looking into including RMS-11K with RSX-11M in the next release.

103. I get the idea that most of the new development is going into RSX-11M-Plus and I appreciate some of the new features. For example, someone put considerable effort into checkpointable commons and they work. But with PDP-11/70's, it is easy to add memory. What I would like to see is memory tricks, such as checkpointable commons, in RSX-11M. We have a big investment in PDP-11/34's and would like to get more mileage out of them. I think ordinary RSX-11M is being abandoned.

We are continuing active development of both RSX-11M and RSX-11M-PLUS. RSX-11M outsells RSX-11M-PLUS by far, and will continue to do so since we do not plan on putting RSX-11M-PLUS on small systems. We will continue to support this very active market for years. However, there are inherent address space limitations within the RSX-11M executive forcing us to be selective about which features are added to both RSX-11 operating systems, or only to RSX-11M-PLUS. Adding features to RSX-11M often reduces pool space, decreasing systme capacities.

104. Closing a file with truncate requires write access to the file. If the file was opened with read-only access, the task hangs.

A patch was published, however the first patch did not properly correct the problem. A second patch has been published.

IAS QUESTION AND ANSWER SESSION
1980 EUROPEAN DECUS SYMPOSIUM

The following is a transcript of the IAS Question and Answer session held at the 1980 European DECUS Symposium in Amsterdam. The transcript was provided by Dr. Otto Titze of the Kernphysik Institut in Darmstadt, West Gerrmany.

1. Is it possible to have a real background task which only exhausts the time left over?

   Start a realtime task and alter priority below the timesharing scheduler. When doing this, alter priority to 2 or greater. Altering priority to 1 will put it below the scheduler's pseudo null task in the ATL so it will never run. Also beware that running a low priority realtime task that uses all available time will effectively stop all batch streams.

2. Could we have a task attribute "realtime" settable at task build?

   We will think about that.

3. Alter priority does not work in a timesharing partition?

   It works on realtime tasks only, not on timesharing tasks.

4. Could DISMOUNT check for installed tasks? If after a dismount, another disk is used, a previously installed task causes problems.

There are no plans for such a feature.

5. Could TKB have a /PR:0 switch for compatibility with RSX-11M. This would allow a privilege task to then be built to 32 KW.

No. A user responded that ZAP could be used to set the privilege bit in the task header.

6. The RK05 diagnostics do not work.

There is no known problem, we will look into it.

7. When will the be a RM02 on-line diagnostic?

The support of on-line diagnostics will not be continued. Use IOX which will come in IAS V3.1.

8. Could you supply some sort of RMDEMO showing the timesharing level tasks currently in memory.

There are no plans.

9. Will IAS run on a LSI 11/23?

No, but it is not clear why not.

10. Will IAS run of a PDP 11/44?

Yes, IAS V3.1 does. It is in field test right now.

11. The elapsed time limit in batch is useless. Could be get real CPU time limits?

Not possible, set no time limit for batch.

12. What happened to the Update kit?

It is no longer on hold and should be available.

13. When booted, SPR is aborted when spooling is set to the systems device.

Ignore the message.

14. Can RSX-11S be generated under IAS in a similar way as RSX-11M.

Yes, there already is a Software Product Description for this.

15. The Guide to Writing Device Handlers manual should include a skeleton driver if in the future new devices are not supported.

There is a new manual for IAS V3.1 with more examples but no skeleton driver. A user responded that the RL01/02 driver is a good example.

16. The CLI manual should be improved, at least it should state CLI's should be multiuser.

OK.

17. Could we have a separate LUN for Login/Logout?

This would be a useful feature and will be considered for the future after IAS V3.1.

18. Could SCI have a subset of low-privilege commands, for example to allow tape mounting?

This sounds useful, probably for the future.

19. When will IAS V3.1 be released?

In early 1981.

20. Will there be problems going from IAS V3.0 to V3.1?

The task headers have changed so all tasks will have to be rebuilt. There is also now a check for the system version in the task header. ATL has changed and there are new manuals.

21. There is always a Software Dispatch article concerning ordering of the PUD to avoid I/O rundown problems. Why don't you reorder the command file for SGN1 which is distributed and solve the problem?

We will consider it.

22. The sysgen command files no longer specify a separate MCR partitions. Is there no need for it?

The MCR partition would still be goods for TKTN to avoid possible system hangup when a task exits or aborts.

23. Why are there free spaces in the DEMO display which are actually not free?

Rebuild DEMO to show actual EXEC, SCOM sizes.

24. TKB does not show fatal errors in the map file, they only come to the user's terminal. Could this be changed to also go to the map?

We will look at this.

25. List directed I/O in F4P V3.0 does not work with a F4PRES. It works fine if the task is linked without one.

Send in an SPR.