RSX-11/IAS SIG DECUS U.S. CHAPTER

THE MULTI-TASKER

Volume 15, Number 4 October 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

TABLE OF CONTENTS

Columns

From the Editor. 104
Eating in L.A. 105
Speak Out. 106
From Five Years Ago. 106
Help Yourself. 107
Hints and Things 110
PDP-11/44 FPP Bug
Building Non-Protected Multi-User RSX-11M
Using TKB's "ACTFIL" Option from Macro-11
Fortran Constants
TTDRV and Control-U
From the Wizards Book of Magic 113
Declaring Non-Standard AST's
Software Performance Reports 118
CODRV/TRACE Error
TKB Error with Resident Commons

Articles

Free Associations. 121
Reading and Writing File Attributes. 128
RSX-11M Announcement 134
PRESERVE, DSC, BRU 142
F11ACP Performance Measurements for RSX-11M. . . 145
A System for Technical Manuscript Preparation. . 160

Special Section

RSX/IAS Sig Tape Abstracts (Part 3). 69

FROM THE EDITOR

This issue has many outstanding articles, starting with Mark Lewis's reflections on the SIG. Mark founded the RSX/IAS Special Interest Group. Much of the success the SIG enjoys is a direct result of Mark's aggressive and far-sighted leadership. About nine months ago, Mark recorded some thoughts on the early days onto tape. A transcription of this tape is this edition's lead article.

The Multi-Tasker has also spared no effort to bring you the latest news from the RSX world. At great personal sacrifice, I attended the European DECUS Symposium in Hamburg where Digital annouced RSX-11M V4.0 and RSX-11M Plus V2.0. In this edition, your roving editor reports on the annoucement.

There are also excellent submissions from Joe Sventek, Dan Dill, and T. Breitschmid. Joe has made extensive F11ACP performance measurements and used Eric Levy's multi-user trick (Multi-Tasker, February 1981) to create a multi-user F11ACP task image. Professor Dill documents tools used at Boston University to support technical manual preparation, including mathematical notation. Mr. Breitschmid has submitted his Fortran-callable subroutines for reading and writing file attributes directly using F11ACP QIO's.

Two new columns debut in this issue and an old one returns. The SIG has a rich history in its past newsletters. I finally have a complete set of back issues, so starting with this issue the Multi-Tasker will carry the best "From Five Years Ago". If this is popular, the column will be expanded to ten years when the time comes. The other new column is titled "Magic". The Multi-Tasker will publish bits and bytes of the past symposium magic shows. Also any new items may be submitted to this column. In general, "Hints and Things" will carry material related to supported systems and "Magic" will have all material which potentially makes your system unsupported. This issue also has the return of the "Speak Out" column. Besides carrying technical articles, I strongly feel that the Multi-Tasker should be a forum for opinions and viewpoints. In the coming months, I will be publishing some of my own personal views and invite anyone to respond to my arguments or submit their own opinions on other issues.

Finally, if any of you would like to work on the Multi-Tasker, now is your chance. My wife has begun work on her masters and no longer has the time to edit copy. Therefore, I need volunteers to edit various columns and articles. These will thankless, dirty jobs. You will have to input and edit submissions on a monthly basis and exchange media with me (either RX01 floppies, 800 or 1600 BPI tapes, or TU58 cartridges). You need enough free time to meet very tight monthly deadlines. If interested, contact me at (314) 694-4252 or at the Los Angeles symposium. There are openings for editors for each of the various columns as well as special assignments: editting selected articles, researching specific topics, or transcribing recordings of symposia sessions.

Ralph Stamerjohn
Multi-Tasker Editor

EATING IN L.A.

In case you are unfamilar with Los Angeles, the Multi-Tasker has drawn up the following list of restaurants in the downtown area (courtesy of Bob Denny). No recommendations for any of the restaurants is implied. Eating in downtown L.A. is (as usual) more expensive than in some other areas, but some of the city's best restaurants are located downtown. Notable are the Chinese, Japanese and Mexican restaurants. You could close down La Fonda at 2AM and open up Vickmans at 3AM.

Vickmans 1228 East 8th St. 622-3852	Early breakfast and lunch only. In downtown produce market. Open 3AM(!) to 3PM.
Pacific Dining Car 1310 W. 6th St. 483-6000	Oldest steak house in L.A. Open 24 hours. Charcoal broiler, seafood also.
La Fonda 2501 Wilshire Blvd. 380-5055	Not far away, Mid-Wilshire. This is it for Mexican food! Unbelievably good till 2AM.
Horikawa 111 S. San Pedro 680-9355	Excellent traditional Japanese. Teppan table grill, sushi bar. In Little Tokyo, closes 10:30 PM.
Italian Kitchen 420 W. 8th St. 622-9277 622-6806	Casual dress, family owned. Reasonable prices, good food. Reservations. Closes 9 PM!
Hong Kong Low 425 Gin Ling Way 628-6217	In Chinatown, one of the best. Wide variety of styles, foods. Closes at 11PM.
Akasaka Hanten 123 S. Weller 617-1100	Szechuan, Shanghai & Peking. Chinese. Reasonalbe prices. In Little Tokyo. Closes at 10PM.
El Cholo 1121 S. Western 724-2773	54 yrs, Cuervo 1800 margaritas. Quite good, but closes at 10PM. Garden/Hacienda atmosphere.
Boston Half Shell 3455 Wilshire Blvd. 487-0177	For those who can't stay away from Mass. Fresh seafood, lunch dinner till midnight.
Nipa Hut 326 S. Alvarado 483-8988	Phillipine Cuisine. Unusual but good. Casual dress, reasonable prices. Closes at 10PM.

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.

SPEAKING OUT?

Ralph Stamerjohn
Multi-Tasker Editor

One of the first things you notice when reading the early editions of the Multi-Tasker is that the pioneer SIG members were very outspoken. If Digital adopted an unfavorable policy, or DECUS moved contrary to the general membership, or the SIG leadership scheduled a bad symposium, you read about it in the Multi-Tasker.

Somewhere, this free exchange of opinions has been lost. I know people still have their views because every symposium, I hear them in the product panels, question and answer sessions, and SIG general meetings. I sometimes grow tired of complaining and listening to complaints about the same issues. Maybe it is time to start exploring whatever power the written word may have.

If you question a Digital policy, send an open letter to the product management and the Multi-Tasker. And then publish the response you receive. If you find that symposiums do not meet the new-user needs, write a critique to the Multi-Tasker. If you feel the Multi-Tasker's pages are too valuable for this type of input, write an editorial. If no one rebuts you, the column goes.

Do not forget that praise is also a powerful persuasion tool. But most importantly, voice your opinions.

FROM FIVE YEARS AGO

The major technical submission in the October 1976 issue of the Multi-Tasker (Vol. 6, No. 4) was a description from Malcolm Gwynne of Bendix Field Engineering on how to setup a hot start using RSX-11D, i.e. a system which would take a crash dump automatically, reboot itself, restore the context of the

application, and then continue. Malcolm's system was used as a host for a satellite computer which is collecting scientific data in a 24 hours/day operational mode. The satellite had a worst case survival time of a little over two minutes when the RSX-11D system crashed. The technique used would still be valid for todays RSX-11M systems:

1. A dedicated area of one disk was allocated for crash dumps. This was done by allocting the area as bad blocks using INI.

2. All variables related to satellite data collection were collected in a global common.

3. The crash module was modified to dump the system to the preallocated space.

4. On restart, programs picked up the saved global common and used the information in it to hot start the applications from before the crash.

The issue also printed 27 Software Performance Reports and an annoucement that Glenn Everhart had submitted a RSX version of Focal to the DECUS library.

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

ACC UMC-Z80

This response was received from Flemming Pedersen of A/S Datalog in response to Art Kocsis question on UMC-Z80 support in the June 1981 issue of the Multi-Tasker. For more information, contact Mr. Pedersen at A/S Datalog, Roskildevej 398, Rodovre, DK-2610 Denmark (Telex DK-15080).

A/S Datalog Denmark has developed telecommunication software for the ACC UMC-Z80, which makes it an intelligent interface for connection of terminals and/or mainframes to the PDP computers.

For readers who do not know of the UMC (Unibus Micro Channel) from ACC Santa Barbara, it is a 4Mhz Z80A based microprocessor board with two serial communication lines, counter timer circuit, DMA access to the Unibus, 4k bytes RAM and room for 16k bytes PROM. An optional TIO board consists of a Z80A CPU

supporting eight serial communication ports, 32k of RAM, 16k of ROM, and counter timer circuit. The UMC and TIO can be connected via an external Zilog bus. Up to four TIO's can be connected to one UMC. Each board mounts in a PDP-11 hex small peripheral controller (SPC) slot.

Following protocols are available:

2780/3780/HASP	IBM batch
NTR	Univac batch
X.25 level 2	HDLC/LAPB following CCITT recommendations
U200/UTS400	Univac interactive applications
3270	IBM interactive applications

The batch protocols use only the UMC board and support dual hosts. The interactive protocols use both types of boards. A typical application involves one or two centralized hosts with DEC minicomputers at field sites. A UMC-system attached to the PDP's will give the users at the sites independent access either to their PDP or the hosts. A few keystrokes on the keyboard will shift the terminal between the computers.

All protocol handling, including character conversion, is done in the frontend processor. Transfer of data between Unibus and UMC is via DMA. This concept significantly lowers the CPU (PDP-11) overhead normally associated with communications.

As a software system house we have used the UMC-TIO system as terminal multiplexor in our in-house system (11/34 with RSX-11M) for about two years now, and systems installed in Germany, UK, Switzerland, The Netherlands, USA, Norway, Sweden and Denmark.

DUAL-PORTED DISK DRIVES

Hans Gunnar Helgesson of the National Defence Research Institute of Sweden responded to M.L. Woods request for help in dual-porting disk drives under RSX-11M.

At our site, we have a dual-ported RM03 connected to a PDP-11/34 and PDP-11/45. Both systems run RSX-11M and have been in heavy use since 1978 with no problems. In our solution, the driver for the RM03 (DRDRV) was slightly modified to support dual access by incorporating a flag which is set if the drive is busy servicing the other port. The drive interrupts DRDRV when it is available, and by checking the flag, a branch is done to continue the pending operation.

Since there is no support for dual-ported disks in Files-11, our solution was to implement complete read/write privileges from on computer (11/34) and only read and update privileges from the other (11/45). This was done by patching the legal function masks in the DCB of the 11/45 system to support read and update operations only. These changes at least let us exchange data between the PDP-11/34 and PDP-11/45.

OTHER QUESTIONS

C.J. Thompson of the Montreal Neurological Institute had an answer for George Velez's problem of reading RT-11 ASCII files under RSX-11M. In general, Mr. Thompson's solutions are the same as outlined last month by Dr. M.G. Liverman. Both Mr. Thompson and Dr. Liverman agree that the easiest method is to copy the files from tape to an RT-11 formatted disk and then from the RT-11 disk to the RSX-11M disk. This is done by "MOUnting" the tape and using the following FLX commands (assume the intermediate disk is a RX02):

```
FLX DY:/RT=MT:file.ext/RS/IM
FLX SY:/RS=DY:file.ext/RT/FA
```

Two additional answeres where received to Wayne Guerrini's problem with linking to resident commons and libraries which are exactly 4KW long. Bob Denny of Creative System Design work-around is to shorten the common area by two words. However, the definitive fix was provided by Robert Thomas. He SPR the problem and received a patch from Digital. This is published in the SPR section of this newsletter.

THIS MONTH'S QUESTIONS

RSX-11M+ DA11-B DRIVER

I would be pleased to hear from anyone that has converted the RSX-11M DA-11B driver (XBDRV) to function under RSX-11M+. If anyone has done this, it would save quite a bit of work.

Dave Evans, Department of Physical Plant, The University of Alberta, Edmonton, Alberta, Canada. T6G 2H1. Phone (403) 432-2986.

RSX-11D SPAWN SUPPORT

We are an RSX-11D site with some software that contains SPAWN. We would like to use the software, but can't figure out what to replace SPAWN with. Can anyone help?

Randolph P. Brown, Computer Sciences Corporation, 9311 Groh Road, Grosse Ile, Michigan, 48138. Phone (313) 226-7811.

LPS-11 AND ICS SUPPORT

I am installing a LPS-11 and ICS into a PDP-11/34 and have no documentation except the I/O drivers manual. I am interested in talking to anyone about these devices and getting examples of how to use them in Macro-11 or Fortran.

Micheal Beland, The Continental Group, 18 Continental Blvd., Merrimack, New Hampshire, 03054. Phone (603) 424-4162.

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.

PDP-11/44 FPP BUG

The following article is from K.D. Hafen, McDonnell Douglas Automation Company, 3855 Lakewood Boulevard, Long Beach, California, 90846.

The PDP 11/44 floating point unit, the FP11-F has a microcode bug. It occurs on the floating point instruction:

```
STEXP   ACN,(RN)+
```

As documented this instruction should increment Rn by two. Unfortunately as it works it will increment Rn by four in single precision mode and by eight in double precision mode. Maybe this is why your program works on the PDP 11/70 but fails miserably on the PDP 11/44.

This problem has been brought to DEC's attention, and was verified in Maynard. Presumably there will be a Hardware ECO coming out sometime in the future.

BUILDING NON-PROTECTED MULTI-USER RSX-11M

The following article is from Marg Knox, The University of Texas at Austin, Austin, Texas 78712.

In the Q and A transcript a user requested information on sysgening a multi-user but unprotected system. This was an option in 3.1, but not in 3.2. It is, however, easy to do:

1. Specify "no" to multi-user protection question asked during the SYSGEN Executive section (essentially you are specifying single user at this point).
2. Immediately exit (CTRL-Z) from VMR phase in SYSGEN 2.
3. Manually build multi-user tasks using their build files: MCRMUBLD, HELBLD, BROBLD, SHUTUP, (e.g. TKB @MCRMUBLD).
4. Edit SYSVMR.CMD to delete "INS MCR".

5. Edit SYSVMR.CMD to add installs for all the multiuser tasks: MCRMU, HEL, BRO, SHUTUP, (e.g. INS MCRMU).

6. Continue with VMR process.

When all done you will have a multi-user system that supports HELP, BROadcast, and SHUTUP. HELLO does nothing and I do not bother to build BYE. This setup has worked very nicely for us. Depending on your kit you may need to juggle your UIC for the taskbuild, editing, and re-VMR phases.

USING TKB'S "ACTFIL" OPTION FROM MACRO-11

The following article is from Robert Denny, Creative Systems Design, 3452 East Foothill Boulevard, Suite 901, Pasadena, California, 91107.

One of the more annoying 'features' of FCS is the requirement that the FSR block buffers be declared at assembly time, rather than allocated as needed at run time.

Sometimes it would be nice enough if one could vary the FSR block buffer allocation at taskbuild time. There are procedures given for this in the I/O Operations Manual (RSX V3.2), section 2.7. The procedure for MACRO-11 programs does not allow for reducing the size of the FSR declared at assembly time, while the procedure for FORTRAN does. Also, the MACRO-11 procedure requires computation of the total buffer pool size in bytes, not too convenient in an indirect command file, or for less experienced users.

All you have to do is say somewhere in your MACRO-11 program is (in OCTAL radix!)

```
F.BFHD == 20
```

and the taskbuilder's "ACTFIL=n" option will work, reserving enough space for "n" simultaneously open files, even if "n" is less than the number specified in the "FSRSZ$" call in the MACRO program.

FORTRAN CONSTANTS

The following article is from Joseph E. Prevo Jr., General Physics Corporation, One Northgate Park, Chattanooga, Tennessee, 37415. He points out a common problem with Fortran that generates very hard to find errors.

I was recently developing a Fortran-IV Plus program under RSX-11M when I seemed to be plauged by an invisible bug. After extensive searching, I found my problem was due to compiler optimization and Fortran-IV Plus subroutine calling conventions.

I had passed a 'constant' to a subroutine when I should have passed a variable. In the subroutine, the variable was incremented and upon return, the 'constant' no longer had its original value.

Fortran-IV Plus passes the address of calling arguments to subroutines and functions. So when constants are used as subroutine arguments, the constant is stored in memory and its address passed to the subroutine. However, Fortran-IV Plus, being an optimizing compiler, checks for constants of the same value and uses the same memory occurance whenever the constant is referenced; thus a common variable 'constant' is produced. The program and subroutines below demonstrate this feature:

```
      PROGRAM TEST
      CALL ADDTWO(10)
      CALL PRNTIT(10)
      END

      SUBROUTINE ADDTWO (I)
      I = I+2
      RETURN
      END

      SUBROUTINE PRNTIT (J)
      WRITE  (5,1000) J
1000  FORMAT (' ',I3)
      RETURN
      END
```

The resuling value printed out is 12. This is not a bug, but I feel that this problem may also go unnoticed by others. If you are having mysterious Fortran errors, one thing to check is constants passed in subroutine and function calls.

TTDRV AND CONTROL-U

The following article is from Bruce McCulley and Micheal Zaharee, instructors for Digital Education Services in Bedford, Massachusetts.

We recently had hardware problems in our DH11 which manifested as progressive control-U's. The problem progress until no input could be entered from any terminal. While the problem looked like pool depletion, powering off and on any terminal would cure the problem temporarily. The problem occured on only one processor but on several operating systems, both RSX-11M and RSX-11M Plus and we verified it was not a pool problem.

After digging into the terminal driver code, we discovered that certain hardware errors would also cause a control-U response. In our case, the DH11 was detecting a framing error. The DH11 line clock was apparently "drifting" off frequency and would be reset by powering off and on a terminal.

Readers should beware that the terminal driver may generate a control-U for several possible errors and not just pool depletion. If you get this problem, it could indicate possible hardware problems with your terminal interfaces.

FROM THE WIZARDS 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.

DECLARING NON-STANDARD ASTS

The following article is from Dan B. Curtis, Science Applications Inc., Oak Brook, Illinois.

This article describes how to send non-standard ASTs to tasks. This method can be used in a variety of ways. One is for full duplex drivers that need to wake up a task when unsolicited input occurs. Another is as a variable send data. Also, a privilege task may use the technique to declare AST's for itself or any other task.

The routine and test program are listed later. The routine SNDAST may be called only from system state. You are in system state after a SWSTK$ or FORK. The routine will send an AST of a specified size to a specified task. If the task is checkpointed out of memory, it will be brought into memory and scheduled to run. The task that receives the AST must remove the arguments from the stack and exit the AST in the normal manner. To facilitate the removal of the non-standard parameters, the SNDAST routine places the number of bytes that must be removed from the stack as the first element in the parameters received by the task. This allows the following code sequence to clean the stack and exit the AST:

```
        ADD     (SP),SP         ; CLEAN UP NON-STANDARD
                                ;   AST VARIABLES
        ASTX$S                  ; EXIT AST
```

The TSTAST program is a simple program to test the working of the SNDAST routine. It simply enters system state, sends a line of text to itself in an AST packet, and when the AST occurs, prints the line. This continues till the operator aborts the task. To assemble and task build the programs:

1. Log onto a privileged account.

2. Type in the programs and command file.

3. Execute the command file.

4. Run the program. BUT only on a machine without anyone else on it. You probably made a mistake and will cause a system crash.

5. When it works congratulate yourself, and look at the QASTT routine in module REQSB and the AST exit directive in module DRATX to see how this mechanism works.

```
        .TITLE  SNDAST
        .IDENT  /V1/
;
;
; WRITTEN BY D.B.CURTIS
; SCIENCE APPLICATIONS, INC.
; 28-SEP-81
; ARGONNE NATIONAL LAB
;
; VERSION 01
; EDIT NUMBER = 0001
; FILE = NSTAST.MAC
; EDITED BY: D.B.CURTIS  28-SEP-81 14:51:11
;
; MODIFICATIONS:
;
;+
; FILE DESCRIPTION
;
;       THIS ROUTINE SENDS AN AST TO A SPECIFIED TASK.
;       THE ROUTINE MUST BE CALLED FROM SYSTEM STATE.
;-
;
        .PAGE
;+
;  **-SNDAST-SEND AST TO TASK
;
; THIS ROUTINE SENDS AN AST TO A SPECIFIED TASK
; NOTE! IT DOES NOT CHECK THAT THE PARAMETERS ARE CORRECT.
; INCORRECT CALLS TO THIS ROUTINE WILL CRASH YOUR SYSTEM!
;
; YOU MUST BE IN SYSTEM STATE BEFORE CALLING THIS ROUTINE.
;
; THE AST THAT IS SENT HAS THE NUMBER OF BYTES THAT MUST BE REMOVED
; FROM THE USER STACK AS THE FIRST PARAMETER ON THE STACK.
; THE MAXIMUN NUMBER OF PARAMETERS THAT YOU CAN SEND IS 125(10)
;
; INPUTS:
;       R0 = NUMBER OF USER PARAMITERS TO PUT IN AST
;       R1 = POINTER TO USER PARAMITERS
;       R2 = AST ADDRESS IN RECIEVER TASK
;       R5 = TCB ADDRESS OF TASK TO SEND AST TO
;
; OUTPUTS:
```

```
;       CARRY CLEAR IF NO PROBLEMS, SET IF ERROR
;
; STANDARD REGISTER USAGE
;
;-
;
; ROUTINES CALLED
;
        .GLOBL  $QASTT          ; QUEUE AST TO TASK
        .GLOBL  $ALOCB          ; ALLOCATE CORE BLOCK

        .PAGE
SNDAST::
        JSR     R5,$SAVRG               ;; SAVE REGISTERS R3-R5
        INC     R0                      ;; ADD OUR PARAMETER
        MOV     R0,R4                   ;; SAVE SIZE
        MOV     R1,R3                   ;; AND POINTER TO USER DATA
        MOV     R2,-(SP)                ;; SAVE R2 AS ALOCB MODIFIES IT
        ASL     R0                      ;; MAKE NUMBER OF BYTES FOR USER PARAMS
        ADD     #5*2,R0                 ;; ADD ON THE SYSTEM PARAMS
        MOV     R0,R1                   ;; COPY SIZE TO R1
        CMP     #177,R1                 ;; CHECK SIZE IS LEGAL
        BHIS    4$                      ;; 200 AND MORE IS ILLEGAL SEE
                                        ;; CRASH DUMP ANALYZER MANUAL PAGE B-39
        SEC                             ;; CARRY IS ALREADY SET BUT FOR CLARITY.
..
        BR      5$

4$:     CALL    $ALOCB                  ;; ALLOCATE A CORE BLOCK, SEE
                                        ;;   EXEC MODULE CORAL
                                        ;;  R0 = ADDRESS, R1= SIZE, R2 = WIPED
5$:     MOV     (SP)+,R2                ;; RESTORE R2
        BCC     10$
        RETURN                          ;; ERROR RETURN

10$:    MOV     R0,-(SP)                ;; SAVE ADDRESS OF AST BLOCK
        CLR     (R0)+                   ;; CLEAR THE LINK
        MOV     R1,(R0)+                ;; SAVE THE SIZE
        MOV     R4,R1                   ;; GET NUMBER OF USER PARAMS
        ADD     #7,R1                   ;; ADD NUMBER OF SYSTEM PARAMS
                                        ;;   LOOK AT EXEC MODULE DRATX FOR MAGIC
 7
        ASL     R1                      ;; TIMES 2 TO MAKE BYTES ON USER STACK
        MOV     R1,(R0)+                ;; SAVE NUMBER OF BYTES ON USER STACK
        MOV     R2,(R0)+                ;; SET AST ADDRESS
        MOV     R4,(R0)+                ;; SAVE NUMBER OF USER PARAMITERS
        MOV     R4,R1                   ;; GET NUMBER OF USER PARAMITERS
        ASL     R4                      ;; MAKE SIZE AS BYTES
        MOV     R4,(R0)+                ;; AND MAKE FIRST ARGUMENT ON STACK
        DEC     R1                      ;; HAVE ALREADY SAVED SIZE
11$:    MOV     (R3)+,(R0)+             ;; SAVE USER PARAMITERS
        SOB     R1,11$
        MOV     (SP)+,R1                ;; GET AST BLOCK ADDRESS
        MOV     R5,R0                   ;; GET RECIEVER TASK TCB
        CALLR   $QASTT                  ;; SEND AST AND RETURN, SEE REQSB

        .END
```



```
        .TITLE  TSTAST
        .IDENT  /V1/
;
;
; WRITTEN BY D.B.CURTIS
; SCIENCE APPLICATIONS, INC.
; 29-SEP-81
; ARGONNE NATIONAL LAB
;
; VERSION 01
; EDIT NUMBER = 0001
; FILE = TSTASTA.MAC
; EDITED BY: D.B.CURTIS  29-SEP-81 16:28:05
;
; MODIFICATIONS:
;
;+
; FILE DESCRIPTION
;
;       THIS FILE IS USED TO TEST THE SNDAST ROUTINE.
;       IT IS A PRIVALIGED PROGRAM (TO ALLOW A SWSTK$) WHICH CALLS
;       THE SNDAST ROUTINE WITH A LINE OF TEXT AS THE PARAMETERS.
;       THE AST ROUTINE ATTEMPTS TO PRINT THE LINE WHEN THIS TASK FROM THE
;       STACK. THIS TEST PROGRAM SENDS AN AST TO ITSSELF AND PRINTS THE
;       AST DATA FROM OFF OF ITS STACK.
;
;       REMEMBER, THE SNDAST ROUTINE CAN SEND AN AST TO ANY TASK.
;-
;
;
        .MCALL  QIOW$S,EXIT$S,ASTX$S

        .PAGE
;
; IMPORTAINT GLOBALS
;
        .GLOBL  SNDAST          ; ROUTINE TO SEND ASTS

;
; LOCAL DATA
;
SAVSP:  .BLKW   1               ; VALUE OF STACK AT STARTUP

PRAMS:  .ASCII  /THIS IS A TEST OF THE AST CODE/ ; TEST MESSAGE
BLKSZ=.-PRAMS
        .PAGE

START:
        MOV     SP,SAVSP                ; SAVE STARTUP TIME SP
1$:     SWSTK$  15$                     ;; GO TO SYSTEM STATE

        MOV     $TKTCB,R5               ;; GET OUR TCB ADDRESS
        MOV     #<BLKSZ+1>/2,R0         ;; GENERATE NUMBER OF PARAMETERS
        MOV     #PRAMS,R1               ;; GET POINTER TO USER PARAMETERS
        MOV     #ASTAD,R2               ;; AND AST ADDRESS
```

```
        CALL    SNDAST                  ;; SEND AST AND RETURN
        RETURN

15$:    CMP     SP,SAVSP                ; CHECK THAT SP OK
        BEQ     1$                      ; AND IF SO, CONTINUE FOREVER
        IOT                             ; ELSE CRASH

ASTAD:                                  ;-- AST ENTRY POINT
        .IRPC   REG,<012345>            ; SAVE REGISTERS
        MOV     R'REG,-(SP)
        .ENDM

        MOV     SP,R0                   ; GENERATE POINTER
        ADD     #14,R0                  ; TO LINE
        MOV     (R0)+,R1                ; GET SIZE
        QIOW$S  #IO.WVB,#5,%1,,,,<R0,R1,#40> ; WRITE LINE

        .IRPC   REG,<543210>            ; RESTORE REGISTERS
        MOV     (SP)+,R'REG
        .ENDM

        ADD     (SP),SP                 ; CLEAN STACK
        ASTX$S                          ; EXIT AST


        .END    START


;
; ASTTSTGEN.CMD
;
        .ENABLE SUBSTITUTION
        .OPEN #0 TEMP.CMD
        .ENABLE DATA #0
MAC SNDAST,SNDAST/-SP=LB:[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,SY:'<UIC>'SNDAST
MAC TSTAST,TSTAST/-SP=LB:[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,SY:'<UIC>'TSTAST
        .DISABLE DATA #0
        .CLOSE TEMP.CMD
        @TEMP
        .OPEN TEMP.CMD
        .ENABLE DATA #0
TSTAST/PR:5,TSTAST/-SP=TSTAST,SNDAST
LB:[1,1]EXELIB/LB
LB:'<SYSUIC>'RSX11M.STB/SS
//
        .DISABLE DATA #0
        .CLOSE #0
        TKB @TEMP
        PIP TEMP.CMD;*/DE
/
```


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 CODRV and TRACE was submitted by Rodger Miles (Telemed Cardio-Pulmonary Systems, 2345 Hoffman Estates, Illinois, 60195). The unpublished SPR 11-34979 refered to was published in the January 1981 Multi-Tasker.

With the bug in CODRV that causes output to be lost (see unpublished SPE 11-34979), the TRACE module cannot be used to trace a program if it is printing to the CO: device. Since these are both "Digital supported" modules, it would be nice if at least one of them were fixed. Attached is a patch to TRACE.OBJ that converts its QIO's to QIOW's. This seems to fix the problem.

```
        .TITLE  TRACE
        .IDENT  /02.01/

.BIAS = .

. = .+340
LOOP1:                          ;Label the first QIOW directive
. = .+10
        CMPB    IOSB,#IS.SUC    ;Overlay the wait code with a
        BNE     LOOP1           ;test for CODRV not ready
        NOP

. = .BIAS+712
LOOP2:                          ;Label the second QIOW directive
. = .+10
        CMPB    IOSB,#IS.SUC    ;Overlay the wait code with a
        BNE     LOOP2           ;test for CODRV not ready
        NOP

. = .BIAS+712
        .WORD   6003            ;Change QIO to a QIOW
. = .+6
        .WORD   IOSB            ; and add an I/O status block

. = .BIAS+742
        .WORD   6003            ;Change QIO to a QIOW
```

```
. = .+6
        .WORD   IOSB            ; and add an I/O status block

. = .BIAS+1214
IOSB:   .BLKW   2               ;Add the status block to the end

        .END
```

The following SPR on TKB was submitted by Robert F. Thomas (A.S. Thomas, Inc., 355 Providence Highway, Westwood, Massachuesetts, 02090). This SPR and Digital response solves a problem submitted by Wayne Guerrini in the June 1981 issue of the Multi-Tasker that involved building 4KW resident commons and libraries.

TKB cannot handle shared commons of exact multiples of 4KW when linking to them.

Digital Response:

There was a restriction in TKB which did not allow for extra address checking during situations where shared commons were exactly 4KW or multiples thereof. This restriction can be removed in two ways. The first method requires only that you build your shared region 2 words smaller. Assembling and taskbuilding the region will result in a common region that will map to the entire 4KW (or multiple) since TKB rounds all addresses up to the next higher 32(10) word block boundary.

The second, and more permanent, solution is to apply the following patch to TKB. This will eliminate the problem and also corrects other problems as well.

```
        .TITLE  SGALO
        .IDENT  /25C/
;
; COPYRIGHT (C) 1981
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
; MODIFICATIONS:
;
;       25A     - Fix multi-user privileged task builds
;
;       25B     - Fix window allocation for memory resident
;                 overlays if a null-length segment is specified
;                 first in the .ODL file
;
;       25C     - Fix address caculation for shared commons of
;                 exact multiples of 4kw words
;
        .PSECT

$PC = .

. = $PC+362                     ;25A
        CALL    $PAT1           ;25A
. = $PC+502                     ;25A
        CALL    $PAT2

. = $PC+2560                    ;25B
        JMP     $PAT3           ;25B
        NOP                     ;25B
. = $PC+2566                    ;25B
$RTP3:                          ;25B
. = $PC+2576                    ;25B
RTP15$:                         ;25B
. = $PC+2614                    ;25B
RTP20$:                         ;25B

. = $PC+3762                    ;25C
        JMP     $PAT4           ;25C
. = $PC+4066                    ;25C
OVRFLW:                         ;25C

. = $PC

        .PSECT  $$PATCH

$PAT1:  MOV     R1,S$GROB(R5)       ;Record base virtual address
        SUB     $OFFST,S$GROB(R5)   ;Subtract offset bias from
                                    ; read-only base address
        RETURN                      ;

$PAT2:  MOV     S$GROB(R5),(R3)     ;Set read-only base of library
        ADD     $OFFST,(R3)         ;Include offset bias
        RETURN                      ;

$PAT3:  TST     $LVL                ;At root level?
        BEQ     $J15                ; If EQ - yes
        TST     (R0)                ;Any length to this segment?
        BEQ     $J20                ; If EQ - no, ignore it
        JMP     $RTP3               ;Back to check rest of segments
$J15:   JMP     RTP15$              ;
$J20:   JMP     RTP20$              ;

$PAT4:  BEQ     10$                 ;If EQ - no check for overflow
        BCS     20$                 ; If CS - then overflow occured

10$:    RETURN                      ;
20$:    JMP     OVRFLW              ;

        .PSECT
        .END
```

Assemble the patch and replace the module SGALO in the library TKB.OLB. Use the following as a guide in patching TKB.OLB.

```
>MAC SGALO.POB=SGALO.PAT
>LBR SGALO.OBJ;1=[1,20]TKB.OLB/EX:SGALO
>PAT SGALO.OBJ;2=SGALO.OBJ;1/CS:54162,SGALO.POB/CS:35040
>LBR TKB.OLB/RP/-EP=SGALO.OBJ;2
```

FREE ASSOCIATIONS

Mark Lewis
First RSX/IAS SIG Chairman

One morning in July of 1973 I made two phone calls that resulted in the organization of the RSX-11 Special Interest Group. Why I made the phone calls was to find a way of guaranteeing my own attendance at future symposia. I had first attended a DECUS meeting in the Fall of 1968 as a then brand new user of a LINC-8. The 1968 Fall Symposium was held in San Francisco which was a satisfactory scene since it provided many distractions from an otherwise terribly dull meeting. I next had occasion to attend a DECUS Symposium in the fall of 1971, again in San Francisco. At that time I was still a user of a LINC-8 and by that time John Alderman had formed what had become the OS/8 Special Interest Group. Largely it was the activities of that group and the skill with which Bob Hassinger edited the OS/8 Newsletter that provided the motivation and the inspiration for the initial efforts with the RSX-11 Special Interest Group. My employer, although recognizing the value of DECUS Symposia through the productivity of the laboratory that I ran, nevertheless, could not approve my attendance at meetings unless I was either presenting a paper or chairing a session. And so in 1972 and in the Spring of 1973 I continued to attend DECUS Symposia largely by chairing sessions no one else wanted. When it became obvious in 1973 that I was soon to be a user of an RSX-11D system I call Barbara Kowalczyk in the DECUS office and inquired about PDP-11-based Special Interest Groups. I was somewhat surprised to discover there were none. There were some SIGs that appealed to PDP-11 users on the basis of the application areas that they represented. One that comes most immediately to mind would be EDUSIG which primarily attempted to reach educational users, a world dominated by RSTS users. Having determined through my conversation with Barbara that no PDP-11 mainframe-type committee existed, it occurred to me that it might make sense to organize not one SIG but two, to appeal to the users of what appeared to be the principle operating systems of the system. At that time the PDP-11 basically had three operating systems. Those were DOS-11, an unmitigated horror I won't mention again; RSTS, which had just become RSTS/E as well since it now could run on the PDP-11/45 and take advantage of that processor's extended memory; and RSX-11D. "D" because it was the fourth RSX-11 operating system. The other three, the DOS-based RSX-11A, RSX-11B, and RSX-11C, I never knew anything about.

The second phone call I made that morning was to Bob Branton who had a small PDP-11 running RSTS at Southeastern State College, also in Oklahoma. Bob and I had met at the DECUS Symposium held in Aneaheim in the Fall of 1972. I suggested to Bob that we might want to organize either a single SIG for users of large PDP-11's or two SIGs, one for users of RSX-11D and another for users of RSTS and RSTS/E. Bob thought as I did that the latter idea was more attractive, probably because it would give us both a chance to organize separate SIGs. We decided during that phone call, however, to submit a single abstract for a joint organizational session of the two SIGs. We spent the balance of that phone call drafting the initial prospectus. The initial prospectus simply said we were

going to form two SIGs, one for RSTS, one for RSX-11D and then itemize several goals. Among the goals were exchanging information at symposia, exchanging information via SIG Newsletters, maintaining user files of software problems and user generated fixes, and also serving as welcome wagons for new users.

So, at the San Francisco meeting in the Fall of 1973 we held our joint session. There was alot of skeptism expressed, principly by the older DECUS members who had by then seen alot of efforts to form SIGs of various sorts that died shortly after they were created. Both Bob and I felt the key to survival of a SIG between symposia was a strong SIG Newsletter. We'd both been to a number of DECUS symposia and noted that many people during a session volunteer to do things that seem to be lost once the immediate ambiance of the symposium had faded from memory. I felt that a strong newsletter, one that encouraged user participation largely by printing a lot of user's names would probably be the key to our survival as a viable organization.

The one other obvious fact that was apparent during that meeting was that the users of RSX-11D were not legion when compared with users of RSTS/E. In fact, I would guess less than 20% of the attendees at that first session were interested in RSX-11. One of the principle noise makers at the meeting, standing in the back of the room, was Mel Woolsey, then the product manager for RSX-11D, who made his presence known by continuing to wave his arms and inquire whether he could join too. Since it was obvious that the RSX-11D users were by far the minority of people present in the room, which was over crowded, I suggested that the RSX-11D users adjourn to the exhibit hall, which was largely deserted, where we could exchange names and get organized. Before leaving the room we did circulate a sign up sheet. That sheet contained about forty names and on later investigation about one-third of them proved to be employees of Digital. Further, not all the names on the sheet represented people present at the meeting. One example that comes to mind is Sally Shlaer, who was not present at that symposium but Tom Provost, who was, added her name to the sign up sheet. I met Tom when we adjourned to the exhibit hall along with Mel Woolsey and about a half dozen other RSX-11D users. The ones that come to mind were Bob Mays, Pat Clarke, and Larry McGowan, largely because they took the notes for that meeting. Indeed I would say it was that half dozen people who compiled the original RSX-11D wish list.

Among the more notable events that occurred at that 1973 Fall Symposium was one paper session headed by a user, Eric Pollack, then of the University of Washington, who presented a simplified scheme for implementing timesharing in RSX-11D, a feature that many users seemed to want, and about a three hour question and answer session, one that was not originally scheduled in the program but held in response to user demands, at which Hank Krejci, the principle architect for RSX-11D, fielded questions not only from the users but also from several DEC Software Specialists in attendance.

In January of 1974 I managed to get out the first issue of the RSX-11D SIG Newsletter. The first issue was as you can imagine a compilation of the wish list from the Fall Symposium plus a few other communications I had received either from Mel Woolsey or from the DECUS office and finally a feature titled "Notes From the RSX-11D Underground", in which I printed all the rumors I heard at the Fall Symposium. The principal early rumors were:

1. A product called RSX-11M, touted to be small brother to RSX-11D.

2. A fast compiling Fortran.

3. And a persistent rumor, that a timesharing Cobol was intended for a future release of RSX-11D.

That feature of the newsletter proved to be one of the most popular but one which was also somewhat dangerous since by printing many rumors I chanced the possibility, indeed the probability, that people in Digital would stop talking to me. But it is noteworthy that in our third issue we mentioned the possibility of two new processors involving four-port memories, a CACHE memory, and an additional four bits of address space and amplified our Cobol rumor into a full scale batch timesharing facility to be called BTS. In retrospect, of course, BTS was a rumored precursor of what became IAS.

At the Spring meeting in Boston of 1974, RSX-11M was announced. Indicative of the attitude many Digital people have toward DECUS was the dreadful film-strip that was used in repeated showings for two days to present RSX-11M to the PDP-11 users. The poor fellow who was charged with answering user questions at that presentation, and I don't remember his name, was totally unprepared for the kinds of questions the DECUS members had for him. At one point, Tom Provost asked him if RSX-11M would support BTS. His answer was a definite yes. That Boston 1974 meeting of the RSX-11 SIG (we only had one session) was the first of what was proved to be a long history of nighttime marathon sessions. RSX-11M had been announced. Fortran-IV Plus had been announced and the SIG meeting took the form of a series of questions asked of Mel Woolsey, Bernie Lacroute, who was the product manager for RSX-11M,and Garth Wolfendale, then the managaer for RSX-11D. RUNOFF and TECO had become available although no one knew it because Hank Krejci had stuck them on an undocumented UIC of RSX-11D V4B, the second release. There were more rumors spread, largely precursors of the 11/70, although the basis for those rumors were a pretty revealing set of remarks made by Dick Clayton at one of the sessions. It was at that meeting that I recruited Sally Shlaer to edit the Help Columns of the RSX-11 SIG Newsletter. It was also at that meeting that I met Louis Barton and Bob Douglas, two members of the SIG who were very active during its formative years.

Between the Spring meeting and Fall meeting of 1974 the amount of material I had been receiving for the newsletter had increased dramatically and so starting with the August issue we became a monthly publication rather than bimonthly, as I had originally intended. The thing I most remember about those early issues was the superb set of flowcharts that Bob Douglas prepared from the source listings of the RSX-11D executive. At that time very few sites had sources, they were incredibly overpriced, as were the listings, and Bob poured through those listings and did an excellent job in detailing the action of the executive.

The Fall meeting in San Diego of 1974 introduced Clay Neal as the new RSX-11 product manager and it was Clay's lot to meet the users at a time when the users of RSX-11D became aware of the horror presented by three incompatible Fortrans. Only the earliest users of RSX-11D remember MOP Fortran, an inelegant compiler that produced relatively slow code but a compiler with, for its time, somewhat unique linguistic attributes. Clearly for a system designed for real time applications there was nothing to compare with MOP Fortran. By the Fall of 1974

it had become apparent Digital intended to replace MOP Fortran with two new Fortrans, FOR, a Fortran processor initially delivered with RT-11, and version 1 of RSX-11M and Fortran-IV Plus, intended for the RSX-11 family and those processors with floating point hardware. Fortran-IV Plus retained all of the features of MOP Fortran that appealed to the real time users. FOR had none of them. This left the users of PDP-11/40's which lack the hardware floating point processor required for Fortran-IV Plus somewhat in the lurch since Digital was intending to desupport the only Fortran that provided the features they needed for their real time environment. It was this somewhat hostile reaction that welcomed Clay Neal to the DECUS fold. Indeed it was only after much yelling and screaming, lead by Bob Mangold that Clay finally committed Digital to continue support for MOP Fortran. In fact Clay said MOP Fortran would be around as long as RSX-11D was around. At the time we had no idea the lifetime for RSX-11D was intended to be quite short. Some other features of that Fall 1974 Symposium that I recall were, in addition to another night time wish list session, an attempt by Roger Vossler to inspire the creation of RSX-11 related local user groups: the first of seemingly endless chain of such sessions that lasted, to my knowledge, for four or five straight years without much success. Also during the Fall 1974 Symposium Bill Lennon, then Symposium Chairman for DECUS organized the first symposium committee and invited me to nominate a representative to the committee after first making it quite clear that there was no way he'd tolerate my own presence on that committee.

It was also at that 1974 Fall Meeting that I first became aware that there was a substantial number of users within the RSX-11 SIG who felt quite strongly that the SIG's activities were being limited by seemingly unnecessary restrictions imposed by DECUS and by Digital on the activities of the SIG. I did much to discourage those people since I didn't believe those restrictions offset the loss we would have had in dealing with Digital had we left the DECUS fold. In retrospect, given the growth of the RSX-11 system, I was probably mistaken.

In 1975, I changed the name of the newsletter to the Multi-Tasker and published the first of the questionaires designed to measure user interest in various features of the system. This first mail-in wish-list was largely the creation of Sally Shlaer. The Spring meeting in 1975 was in Miami, a meeting most remembered by the people in attendance for the absence of functional air conditioning in the hotel metting rooms. As a consequence many people attended the meeting but not the sessions or if they did attend the sessions it was only when highly motivated to do so and then only for a short time. It was at that Spring meeting that the SIG steering committee was first formally organized and it was at that session that RSX-11D V6, RSX-11M V2, RSX-11S V1, and IAS V1 were announced. The latter in particular, with its introduction of Digital Command Language and its many incompatibilties with the rest of the RSX-11 family, produced a large vociferous reaction on the part of the RSX-11 users over the lack of compatibility among all members of the family. It was also at that meeting that one of the Digital representatives, Peter Van Roekens, mentioned corporate bounds: he said that there would be "bounds" placed around the development of RSX-11D and of RSX-11M so that these systems did not encroach upon the functionality of IAS. During the Miami meeting Sally presented the results of that first questionaire and commented at one point that the user community seemed to know more about how the users were using the Digital systems than did Digital, a remark that prompted Clay Neal to ask if he could participate in the drafting of our next questionaire. That summer, I, Clay Neal, Al Hulbert, Dale Moore, and Bob Wood met to draft another questionaire. The questionaire was intended to be a priority determination and a problem

reporting mechanism. That is, the group had, in discussions with Clay, determined that wish lists weren't very useful. Clay emphasized that a list of problem areas or of desired features was preferable to a list of suggested solutions, i.e., he wanted the users to present Digital with the problems for Digital's personnel to solve. He did not think it was beneficial for the Digital users to suggest solutions directly. My own feeling was there was some truth to both of those, that is, there were many features that the users wanted. These tended to be the most important features missing from the systems. There are other areas where the users had multiple solutions for individual problem areas and it was in those latter areas that presenting the problems to Digital appeared to make more sense than presenting the solutions.

The Los Angeles Meeting in the Fall of 1975 produced three features of the RSX-11 SIG Symposium profile that have proven to be quite long lived. The first came about when Sally Shlaer organized a set of mini-tutorials presented by various users. At that time, there were many more RSX-11D users in attendance than there were experienced RSX-11M users so she had little difficulty in finding three users to present short mini tutorials for the RSX-11D systems. She was rather less fortunate in locating suitable RSX-11M users, but one experienced user, Eric Baatz, volunteered to answer questions from the audience. Thus was born the question-and-answer session, long a feature of the RSX-11 Symposium. It should be noted the original question-and-answer sessions started out with the users answering questions. Over time as more and more Digital people started attending those sessions they evolved into a format where the users asked questions of Digital rather than of other users. I've always fe t that someday we should at least try a session where Digital asks the questions and the users answer them. At one point during that meeting a group of users were gathered chatting in the lobby outside the exhibit hall when Sally Shlaer was inspired to grab a piece of cardboard and scrawl a sign reading "RSX 11 spoken here" and to tape it up to a column in the lobby. That proved to be eminently successful and has sinced evolved into the SIG campgrounds, now a feature of all symposia. Another permanent feature of symposia was also originated when, noting that I was getting more and more hopelessly confused trying to moderate a session with questions coming from every corner of a large meeting room with about 400 people present, Angela Cossette came up to me and suggested that I have people line up at the microphone. This worked so well that we've used it ever since. The Los Angeles meeting also saw two rumors circulating pertaining to new products, one called Star, another called Comet. It was also at that meeting that Clay Neal in response to increased user demands for support of I & D space, principly in RSX-11D, responded that he felt the obvious solution to the address space problem lay in the design of a new processor rather than in implementation of I & D space.

The real bomb of the Fall 1975 meeting was the dropping of the other shoe: the "stabilization" of RSX-11D (hinted at Miami) was announced and users of RSX-11D were told to expect no further significant development efforts. Migration paths (that satisfied no one) were outlined to push RSX-11D users into the RSX-11M or IAS camps, where development woudl continue. Today, IAS users with RSX-11D origins have the dubious distinction of being the only users of a system to have been stabilized twice!

In the Spring of 1976 we met in Atlanta. Clay Neal had left the company and Carl Gibson became the third product manager for RSX-11. Again the users made noises about Fortran compatibility, noises that were largely ignored, and there were many complaints over the number of bugs in the latest release of RSX-11D,

complaints that resulted during the meeting in Digital creating a hotline for users of RSX-11D and a special effort to make sure the next release, Version 6.2, would be as bug free as possible. It was also at this meeting that the SIG determined to create a set of formal bylaws and the task of drafting the initial bylaws was assigned to June Baker. In this effort, we were somewhat inspired by the example set by the DECsystem 10/20 Group, to base our formal structure on participation by local users groups. The success of the DEC 10/20 Group in basing their structure on LUGs was not emulated in the resulting by-laws for the RSX-11 Special Interest Group largely because there were so few large and well organized RSX-11 LUGs.

In the Fall of 1976 the Symposium was held in Las Vegas and it was at that symposium that Jeff Gallup and Sally Shlaer presented their formal report on incompatibilties between the RSX-11M and RSX-11D families. That formal report and the reply by Digital six months later was the only example in the RSX-11 SIG that I can think of in which our activities mimiced those of the traditional user groups. The fact that DECUS did not appear to be the sort of formal organization that as are SHARE and COMMON is probably largely responsible for the fact that most users present ignored Herb Grosch's remarks in his keynote address at the Las Vegas Symposium. As I recall Grosch's major argument was that user groups existed to "screw the manufacturer before the manufacturer screwed you." In retrospect there was probably alot of truth in Herb's remarks but the absence of a formal adversary relationship between users and Digital caused his remoarks to be belittled or ignored. It was also at Las Vegas that a small group of RSX-11 users met offline with representatives of Digital's Marketing Group to discuss the incompatibilities remaining between Fortran IV and Fortran IV Plus. Of note, the Digital people present all agreed that there was no excuse for the existing incompatibilities but the SIG was totally unsuccessful in influencing the marketing people present who took the position that it didn't make any difference as far as sales were concerned and that the arguments based upon a sites were largely specious.

The following Spring we met in Boston and Digital gave its formal response to the RSX-11M and RSX-11D incompatibility report. A response that was both promising and disappointing. Promising in the sense that they were now aware of the kinds of problems the incompatibilities presented and that they had resolved to solve many of them by fixing RSX-11M, but also extremely disappointing in that the apparent egotistical attitude of some of the Digital people created the impression that they as developers were in a better position than the users to tell the users what the users needed. It was also at the Boston Meeting, because John Barr had brought a copy of Pascal with him, that an ad hoc tape copying facility was created later to find fruition at the following meeting with the establishment of a formal tape copying mechanism. In this regard, the RSX-11 SIG efforts could hardly be called pioneering since I recall as early as 1970 and 1971 the OS/8 Special Interest Group spent many hours copying tapes on the PDP-8 and PDP-12 then brought to the meetings.

The San Diego meeting in the Fall of 1977 was largely uneventful. It was at that meeting that we had the formal tape copying activity. It was at that meeting that the RSX-11 SIG was given a formal campground -- this in response to the demands that Sally Shlaer had made upon DECUS because of the failure of the "RSX-11 spoken here" signs at the Atlanta, Las Vegas and Boston meetings. So at last in San Diego 1977 the oldest established permanent floating crap game in DECUS was given a room of its own. This proved so successful that by the following meeting every SIG had a formal campground. It was also at the San

Diego meeting of 1977 that I resigned as chairman of the RSX-11 Special Interest Group so I can't say too much more about the activities of the SIG in future years. I was a participant in many of the panels held in Chicago, San Francisco and New Orleans and I was pleased to see the SIG hold the first user sponsored and user taught presymposium seminars, a topic we had discussed for many years but only achieved once we met in New Orleans and that only because of the efforts by Mary Ann Esfandiari who had joined the SIG at the Boston meeting.

The San Diego meeting was also eventful in that it was at that meeting the VAX was formally announced and the VAX/VMS SIG was formed. In spite of the best of intentions by both Digital and the users it became readily apparent that the future development of the RSX-11 family was going to be considerably reduced. That a new product, RSX-11M Plus was also announced did nothing to overcome the apparent pessimism that one could see not only in the RSX-11 room but also in the meeting rooms of the RSTS and RT-11 SIGs. I suspect, however, that it was the experience that Digital had and the great amounts of user unrest and dissatisfaction that Digital enjoyed during the early years of the RSX-11 and RSTS Special Interest Groups that prompted Digital to at least attempt with the 32-bit product not to repeat the mistakes made with the proliferation of operating systems that occurred with the PDP-11. I've left out of this history some remarks on the internal feuding in Digital between the developers of RSX-11D (and later IAS) and the developers of RSX-11M and RSX-11M Plus. It goes without saying that feud did nothing to help the user community and did much to hurt them. Many of the incompatibilities between the RSX-11D, IAS architecture and the RSX-11M, RSX-11M-Plus architecture are a direct result of that somewhat unprofessional feud. The incompatibilities that resulted from several different Fortrans can only be regarded as one of the stupidest things that any organization of any size has ever perpetrated. That Digital was experiencing major problems with growth at the time seems largely irrelevant. That Digital learned from its mistakes is quite promising. The SIG also had growing pains during that period and I'm not certain that the Special Interest Group did learn from its mistakes.

It's now a week later and I've just listened to this tape and noticed several errors, largely of omission. I thought I'd ought to at least correct those. Somehow I've left out tne name of John Eagleston, who was a very active user, one of the original founding members, if you wish, back in 1973 who remained active well into 1975. At the Boston 1974 meeting when mentioning the Digital panel somehow the name of Dave Cutler, principal architect of RSX-11M was omitted. In the Spring of 1975, actually the Fall of 1974, when we first had a SIG Symposium Coordinator Roger Vossler was our first representative to the symposium committee. Garth Wolfendale, who vanished from Digital soon after the Boston meeting, re-apperaed a few years later as the Multi-Tasker's Australian editor. A thinly disguised account of his experiences at Digital, including a skeleton sketch of the RSX-11M, RSX-11D feud, was published as part of his paper "Software Engineering", Canberra Computing Bulletin, December 1976.

READING AND WRITING FILE ATTRIBUTES

T. Breitschmid

OWL AARAU
Arrau, Switzerland

We use PDP-11's for warehouse management. The computers must talk to each other and DECnet is out of the question because of its resource requirements. We have developed our own communications package that allows MCR commands to be started on the remote computer, transfer database transaction data, and copy files of any size or type to the other computer.

The ability to transfer any file caused a problem, because besides the actual file, the attributes must be transfered. However, using F11ACP, it is possible to determine the attributes of a file and to open an equvalent file with this data. The two central subroutines used to get the file attributes (GETATR) and create a file with these attributes (PUTATR) are listed below. These routines are Fortran callable and might be useful for a wide number of applications.

```
        .TITLE  GETATR
        .IDENT  /810108/
;
; This subroutine will get a file's attributes and access the file
; for reading. The routine assumes logical unit 3 and uses whatever
; device is assigned to that unit.
;
; FORTRAM-call:
;
;       CALL GETATR (UIC1,UIC2,FILNAM,VERS,BUFFER,SBSTAT,IOSTAT)
;
;               BYTE    UIC1(3)         UIC-group, 3 ASCII-characters
;               BYTE    UIC2(3)         UIC-owner, 3 ASCII-characters
;               BYTE    FILNAM(12)      Filename and filetype, 12 ASCII
;                                       characters. Name must be aligned
;                                       in front and type in back.
;               INTEGER VERSION         File-version number
;               BYTE    BUFFER(46)      Returned file attributes
;               INTEGER SBSTAT          Subroutine status (1 = success)
;               INTEGER IOSTAT(2)       I/O status
;
        .MCALL  QIOW$,DIR$
;
; Macro for testing I/O status.
;
        .MACRO  TSTISB  X,Y
        CMPB    #IS.SUC&377,X
        BNE     Y
```

```
        .ENDM   TSTISB
;
; Definitions.
;
LUN     = 3
FLAG    = 3
;
; QIO definitions.
;
FNAQ1:  QIOW$   IO.FNA,LUN,FLAG,,IOSTAT,,<,,,,,DIRBLK>
FNAQ2:  QIOW$   IO.FNA,LUN,FLAG,,IOSTAT,,<,,,,,FILBLK>
RATQIO: QIOW$   IO.RAT,LUN,FLAG,,IOSTAT,,<FILBLK,ATTRIB>
ACRQIO: QIOW$   IO.ACR,LUN,FLAG,,IOSTAT,,<FILBLK,,,,100000>
;
; Scratch space.
;
IOSTAT: .BLKW   2                       ;I/O status for QIO's

ATTRIB: .BYTE   -1,4                    ;Read file owner, protection
        .WORD   BUF                     ; into 'BUF'
        .BYTE   -5,12                   ;Read filename, type, version
        .WORD   BUF+4                   ; into 'BUF+4'
        .BYTE   -4,40                   ;Read file attributes
        .WORD   BUF+14.                 ; into 'BUF+14.'
        .WORD   0                       ;

BUF:    .WORD   0                       ;Fileowner UIC
        .WORD   0                       ;File protection
        .WORD   0,0,0                   ;File name
        .WORD   0                       ;File type
        .WORD   0                       ;File version
        .WORD   0                       ;Record type
        .WORD   0                       ;Record attributes
        .WORD   0                       ;Record size
        .WORD   0,0                     ;File size (blocks)
        .WORD   0,0                     ;EOF block number
        .WORD   0                       ;First free byte in EOF
        .BLKW   9.                      ;Dummy space
;
; Directory-block for the file.
;
FILBLK: .BLKB   6                       ;File-ID
FILN:   .BLKB   10                      ;File name
FILD:   .BLKB   10                      ;File directory
FILC:   .WORD   "SY                     ;File device
FILU:   .WORD   0                       ;File unit
;
; Directory-block for the directory.
;
DIRBLK: .BLKB   6                       ;File-ID
DIRN:   .BLKW   2                       ;Directory name
        .WORD   0                       ;
        .RAD50  /DIR/                   ; .DIR type
        .WORD   1                       ; version
DIRD:   .BLKB   4                       ;
```



```
        .WORD   4,4,0                   ;Master directory-ID
DIRC:   .WORD   "SY                     ;Directory device
DIRU:   .WORD   0                       ;Directory unit
;
; Start of routine.
;
GETATR::                                ;Ref. label
        MOV     6(R5),R0                ;Get filename
        MOV     #FILBLK+N.FNAM,R3       ;Get place to store
10$:    CLR     R1                      ;Do not convert periods
        CALL    $CAT5B                  ;Convert to RAD50
        MOV     R1,(R3)+                ;Store converted value
        SOB     R4,10$                  ; and loop till done
        MOV     @8.(R5),FILBLK+N.FVER   ;Store version number
        MOV     2(R5),R0                ;Get UIC group number string
        CLR     R1                      ;Do not convert periods
        CALL    $CAT5B                  ;Convert to RAD50
        MOV     R1,DIRN                 ;Store name of directory file
        MOV     4(R5),R0                ;Get UIC owner number string
        CLR     R1                      ;Do not convert periods
        CALL    $CAT5B                  ;Convert to RAD50
        MOV     R1,DIRN+2               ;Store name of directory file

        DIR$    #FNAQ1,ERR1             ;Get file-ID of directory
        TSTISB  IOSTAT,ERR1             ; exit on error
        MOV     DIRBLK+F.FID+0,FILBLK+N.DID+0   ;Store directory-ID
        MOV     DIRBLK+F.FID+2,FILBLK+N.DID+2   ; ...
        DIR$    #FNAQ2,ERR2             ;Get file-ID of file
        TSTISB  IOSTAT,ERR2             ; exit on error
        DIR$    #RATQIO,ERR3            ;Read file attributes
        TSTISB  IOSTAT,ERR3             ; exit on error

        MOV     10.(R5),R1              ;Get buffer pointer
        MOV     #BUF,R0                 ;Get attribute buffer
        MOV     #23,R2                  ;Set loop counter
20$:    MOV     (R0)+,(R1)+             ;Copy attributes to return buffer
        SOB     R2,20$                  ; and loop till done

        DIR$    #ACRQIO,ERR4            ;Access the file for read
        TSTISB  IOSTAT,ERR4             ; exit on error
        MOV     #1,@12.(R5)             ;Return success
        BR      END                     ; and continue

ERR1:   MOV     #2,@12.(R5)             ;Set error 2 (directory QIO failed)
        BR      END
ERR2:   MOV     #3,@12.(R5)             ;Set error 3 (file lookup failed)
        BR      END
ERR3:   MOV     #4,@12.(R5)             ;Set error 4 (read attribute failed)
        BR      END
ERR4:   MOV     #5,@12.(R5)             ;Set error 5 (access for read failed)
END:    MOV     14.(R5),R1              ;Get I/O status block
        MOV     IOSTAT+0,(R1)+          ;Return I/O status
        MOV     IOSTAT+2,(R1)+          ; ...
        RETURN                          ;Return to caller
        .END
```

```
        .TITLE  PUTATR
        .IDENT  /810720/
;
; This subroutine will create the named file with the supplie attributes
; and open the file for writing. Lun 3 is assumed to be assigned to the
; desired device. The filename is taken from the attribute buffer.
;
; FORTRAM-call:
;
;       CALL PUTATR (UIC1,UIC2,BUFFER,SBSTAT,IOSTAT,SUPSED,FILEID)
;
;               BYTE    UIC1(3)         UIC-group, 3 ASCII-characters
;               BYTE    UIC2(3)         UIC-owner, 3 ASCII-characters
;               BYTE    BUFFER(46)      File attributes
;               INTEGER SBSTAT          Subroutine status (1 = success)
;               INTEGER IOSTAT(2)       I/O status
;               INTEGER SUPSED          Supercede exiting file (0=no, 1=yes)
;               INTEGER FILEID(2)       New file-ID
;
        .MCALL  QIOW$,DIR$
;
; Macro for testing I/O status.
;
        .MACRO  TSTISB  X,Y
        CMPB    #IS.SUC&377,X
        BNE     Y
        .ENDM   TSTISB
;
; Definitions.
;
LUN     = 3
FLAG    = 3
FILTYP  = 102400                ;File-type
                                ; 100000: non-contiguous
                                ; 102400: contiguous
;
; QIO definitions.
;
FNAQ1:  QIOW$   IO.FNA,LUN,FLAG,,IOSTAT,,<,,,,,DIRBLK>
FNAQ2:  QIOW$   IO.FNA,LUN,FLAG,,IOSTAT,,<,,,,,FILBLK>
WATQIO: QIOW$   IO.WAT,LUN,FLAG,,IOSTAT,,<FILBLK,ATTRIB>
ACWQIO: QIOW$   IO.ACW,LUN,FLAG,,IOSTAT,,<FILBLK,,,,100000>
ENAQIO: QIOW$   IO.ENA,LUN,FLAG,,IOSTAT,,<,,,,,FILBLK>
CREQIO: QIOW$   IO.CRE,LUN,FLAG,,IOSTAT,,<FILBLK,ATTRIB,FILTYP,LENL>
;
; Scratch space.
;
IOSTAT: .BLKW   2                       ;I/O status for QIO's

ATTRIB: .BYTE   -1,4                    ;Read file owner, protection
        .WORD   BUF                     ; into 'BUF'
        .BYTE   -5,12                   ;Read filename, type, version
        .WORD   BUF+4                   ; into 'BUF+4'
        .BYTE   -4,40                   ;Read file attributes
        .WORD   BUF+14.                 ; into 'BUF+14.'
```



```
        .WORD   0                       ;

BUF:    .WORD   0                       ;Fileowner UIC
        .WORD   0                       ;File protection
        .WORD   0,0,0                   ;File name
        .WORD   0                       ;File type
        .WORD   0                       ;File version
        .WORD   0                       ;Record type
        .WORD   0                       ;Record attributes
        .WORD   0                       ;Record size
        .WORD   0,0                     ;File size (blocks)
        .WORD   0,0                     ;EOF block number
        .WORD   0                       ;First free byte in EOF
        .BLKW   9.                      ;Dummy space
;
; Directory-block for the file.
;
FILBLK: .BLKB   6                       ;File-ID
FILN:   .BLKB   10                      ;File name
FILD:   .BLKB   10                      ;File directory
FILC:   .WORD   "SY                     ;File device
FILU:   .WORD   0                       ;File unit
;
; Directory-block for the directory.
;
DIRBLK: .BLKB   6                       ;File-ID
DIRN:   .BLKW   2                       ;Directory name
        .WORD   0                       ;
        .RAD50  /DIR/                   ; .DIR type
        .WORD   1                       ; version
DIRD:   .BLKB   4                       ;
        .WORD   4,4,0                   ;Master directory-ID
DIRC:   .WORD   "SY                     ;Directory device
DIRU:   .WORD   0                       ;Directory unit
;
; Start of routine.
;
PUTATR::                                ;Ref. label
        MOV     2(R5),R0                ;Get UIC group number string
        CLR     R1                      ;Do not convert periods
        CALL    $CAT5B                  ;Convert to RAD50
        MOV     R1,DIRN                 ;Store name of directory file
        MOV     4(R5),R0                ;Get UIC owner number string
        CLR     R1                      ;Do not convert periods
        CALL    $CAT5B                  ;Convert to RAD50
        MOV     R1,DIRN+2               ;Store name of directory file

        MOV     6(R5),R0                ;Get supplied attribute buffer
        MOV     #BUF,R1                 ;Get buffer area
        MOV     #23.,R2                 ;Get number of words to move
10$:    MOV     (R0)+,(R1)+             ;Copy file attributes
        SOB     R2,10$                  ; and loop till finished

        MOV     #BUF+4,R0               ;Get filename part of attributes
        MOV     #FILN,R1                ;Get filename block
```

```
        MOV     #4,R2                   ;Get number of words to move
20$:    MOV     (R0)+,(R1)+             ;Copy filename
        SIB     R2,20$                  ; and loop till finished

        TST     @12.(R5)                ;Test is supersesding
        BEQ     50$                     ; No, create new file

        DIR$    #FNAQ1,ERR5             ;Get file-ID of directory
        TSTISB  IOSTAT,ERR5             ; exit on error
        MOV     DIRBLK+F.FID+0,FILBLK+N.DID+0   ;Store directory-ID
        MOV     DIRBLK+F.FID+2,FILBLK+N.DID+2   ; ...
        DIR$    #FNAQ2,ERR6             ;Get file-ID of file
        TSTISB  IOSTAT,ERR6             ; exit on error
        DIR$    #WATQIO,ERR7            ;Write file attributes
        TSTISB  IOSTAT,ERR7             ; exit on error
        BR      60$                     ;And continue

50$:    DIR$    #CREQIO,ERR1            ;Create new file
        TSTISB  IOSTAT,ERR1             ; exit on error
        DIR$    #FNAQ1,ERR2             ;Get file-ID of directory
        TSTISB  IOSTAT,ERR2             ; exit on error
        MOV     DIRBLK+F.FID+0,FILBLK+N.DID+0   ;Store directory-ID
        MOV     DIRBLK+F.FID+2,FILBLK+N.DID+2   ; ...
        DIR$    #ENAQIO,ERR3            ;Enter new filename in directory
        TSTISB  IOSTAT,ERR3             ; exit on error

60$:    DIR$    #ACWQIO,ERR4            ;Access the file for write
        TSTISB  IOSTAT,ERR4             ; exit on error
        MOV     14.(R5),R1              ;Get address to return file-id
        MOV     FILBLK+N.FID+0,(R1)+    ;Return file-id
        MOV     FILBLK+N.FID+2,(R1)+    ; ...
        MOV     #1,@8.(R5)              ;Return success
        BR      END                     ; and continue

ERR1:   MOV     #2,@8.(R5)              ;Set error 2 (create failed)
        BR      END
ERR2:   MOV     #3,@8.(R5)              ;Set error 3 (directory lookup failed)
        BR      END
ERR3:   MOV     #4,@8.(R5)              ;Set error 4 (enter name failed)
        BR      END
ERR4:   MOV     #5,@8.(R5)              ;Set error 5 (access for write failed)
        BR      END
ERR5:   MOV     #6,@8.(R5)              ;Set error 6 (directory look failed)
        BR      END
ERR6:   MOV     #7,@8.(R5)              ;Set error 7 (file lookup failed)
        BR      END
ERR7:   MOV     #8.,8.(R5)              ;Set error 8 (write attributes failed)
END:    MOV     10.(R5),R1              ;Get I/O status block
        MOV     IOSTAT+0,(R1)+          ;Return I/O status
        MOV     IOSTAT+2,(R1)+          ; ...
        RETURN                          ;Return to caller
        .END
```


RSX-11M ANNOUNCEMENT

Colin Mercer
European RSX/IAS SIG Chairman

Ralph Stamerjohn
Multi-Tasker Editor

At tne Decus Europe Hambug '81 Conference, Steve Paavola gave the first presentation of the newly announced releases of RSX-11M V4.0, RSX-11S V4.0, and RSX-11M Plus V2.0. Shipment was expected in the beginning part of 1982. The following notes were made from the slide presentation together with observations from the demonstration system running in the exhibit hall.

The new releases make the systems easier to use, provide higher performance, and implement some new features and tools. As you will see from the comments below, the new releases also implement many of the little features long requested by users.

NOTE

From the Editor This article was compiled from a submission by Colin Mercer and my own personal notes. The question and answer session notes are especially sketchy. Please take the information with a grain of salt. This is not, by any means, an exhaustive list of the new features of RSX-11M V4.0 or RSX-11M Plus V2.0.

1.0 DCL

The main new feature from the users point of view is DCL for RSX-11M. DCL stands for Digital Command Language and is a subset of the VAX/VMS DCL. The same implementation is used for both RSX-11M and RSX-11M Plus and is table driven. The implementation includes an imbedded help facility and the ability to add or change DCL syntax. A BNF-style notation is used to generate the DCL parsing tables and Digital stated documentation will be provided to let users add commands.

2.0 USER-WRITTEN CLI'S

RSX-11M V4.0 and RSX-11M Plus V2.0 also implement the concept of user-written CLI's (command line interpreters). The system manager can assign a CLI to each terminal. Digital supplies MCR and DCL, however, support is provided in the executive for user's to write their own CLI's. The CLI assigned to a terminal receives all unsolicited command lines, which it can then can act on as it pleases. This gives you total control of the user interface. It was observed by a user that the SIG tapes a year from now will be full of CLI's implementing UNIX, RSTS, DG and who knows what other command languages.

3.0 SYSTEM GENERATION

SYSGEN is again made easier, but slightly longer as the task build command files for all utilities are generated by the SYSGEN command files. The saved answer files now extend into Phase 2 so a complete system generation can be done unattended. Note, the saved answer files from RSX-11M V3.2 are not compatible but it is possible to generate RSX-11M V4.0 using a RSX-11M V3.2 system. As previous stated, a 28Kw minimum system is now required for a RSX-11M system generation. Digital will distribute a "fully functional" default executive. Another interesting feature is a autoconfigure program which will look for CSR and vector addresses for all standard devices. Taken together, is should be possible to get a working RSX-11M V4.0 system without any system generation at all. Because SYSGEN generates the utlility task build command files on the fly, the size of task images will be correct. More importantly, resident library support for FCS (FCSRES) can be selected and all utilities will be built using this.

System generation for RSX-11M Plus V2.0 has been improved even more. To start with, loadable drivers with loadable data bases are supported. Also, the resident FCS library can be a supervisor mode library. Finally, RSX-11M Plus will have a special distribution kit on one RL02 that will require no system generation. This will be a fully functional executive and loadable drivers with loadable data bases. No sources will be provided, so maintenance will be with update kits only.

4.0 NEW HARDWARE SUPPORT

The major new hardware announcement was support in RSX-11M Plus V2.0 for the PDP-11/24! In addition, RSX-11M Plus will now support the K-series logic and the KMC-11 line printer. RSX-11M V4.0 gains support for the RM05 disk drive and both systems support the ML11 "non-rotating electronic disk" (it looks like a RS04) and the RA80 disk drive. The later was another new annoucement. The RA80 is a Unibus-only 121MB fixed media Winchester drive with a 25 ms average access time. Up to three drives can be placed in one cabinet for a total of 363 MB of storage in 0.4 square meters. The RA80 is interfaces using the UDA (Unibus disk adapter). This is a smart controller that supports overlapped seeks, seek optimization, and dynamic bad block handling!

5.0 TERMINAL DRIVER FEATURES

The full-duplex terminal driver has some new features, especially in the area of dial-up lines. It now supports separate answer speeds for each line and autobaud detection from 110 to 1200 baud. In addition, a new QIO function tells the terminal driver to hang-up a dial-up line. The terminal driver and executive have also been modified to support asynchronous, buffered I/O. This means a task will no longer stop if it issues a QIO without wait to a terminal. Instead, it may continue processing and will be stopped when it finally issues a wait request.

6.0 POOL IMPROVEMENTS

Pool also receives substantial improvements. For RSX-11M V4.0 only, an optional directive common block will free from 4-6 KW's of pool space. The directive common block moves the directive processing code out of the 20K executive and into two new partitions (EXCOM1 and EXCOM2), therby releasing the space for pool. The executive will map EXCOM1 and EXCOM2 automatically when it needs to process a directive. In addition, the full-duplex terminal driver data structures are reworked to use less pool space and the task loader will no longer hang the system if it detects a pool allocation failure. Finally, the executive can support pool usage limits. Two limits are setable with MCR commands: a warning limit and a fatal limit. When the limits are reached, the executive will request a special pool task, which can be user-written. The version distributed by Digital will broadcast a warning message to all terminals when the warning limit is reached. For the fatal limit, Digital's task will take control of the system, output an active task list to the console, and then allow the manager to selectively abort tasks.

For RSX-11M Plus, many new structures are moved to the secondary pool: terminal UCB extension, command line buffers, TCB's of MCR task not currently active, and type-ahead buffers. In addition, RSX-11M Plus keeps the task headers with the task so they no longer contribute to pool usage.

7.0 EXECUTIVE FEATURES

Both RSX-11M V4.0 and RSX-11M Plus V2.0 have new executive features. The asynchronous buffering previously mentioned is available for other user-written drivers to use. At last, the executive will optionally use the EIS instructions. This particularly helps the disk drivers compute cynclinder, track, sector address from block numbers. The shuffler algorithm is improved. Context switching of the FPP registers is now governed by the running system. These means executives built with FPP support will run on non-FPP machines. New directives include:

* Enhanced stop-bit directives so group-global event flags have the same functionality as other flags.

* Lock/unlock group-global event flag directives allow complete control of when group-global event flags are destroyed. This eliminates race conditions that existed in RSX-11M V3.2.

* A requested-exit-AST directive allows any task to receive an AST whenever it is requested to exit, including MCR aborts! This allows a task to clean-up any neccessary I/O or communications. For privilege tasks, they can be setup so they never exit, no matter how many times they are aborted.

* A set-system-time directive allows a privilege task to set the system time and day.

* Messages can be sent from tasks to the error logger with a new directive.

* A new parent/offspring directive allows a offspring task to pass its parent information to its own offspring.

* Four new directives are added for support of CLI's.

8.0 FILE SYSTEM

F11ACP has been enhanced to allow 65,000 files per volume (on large disks) only. Such disks are not compatible with previous releases of RSX-11M. Also, only command files are being distributed for only four different styles of F11ACP tasks so keep your RSX-11M V3.2 versions in case you want them.

9.0 MAGTAPE SUPPORT

Another major feature of the releases in vastly improved support for magtapes. MTAACP has been rewritten to fully support ANSI Level 3 standard, except for user labels. It will handle IBM format tapes and provide EBCDIC/ASCII conversion! A new utility, MAG, is provided. MAG provides many of the basic tape manipulation features like rewind, space file, and so on.

10.0 ERROR LOGGING

The error logging package has also been rewritten completely. This was done in cooperation with Field Service and the new reports are reported to be much more extensive and complete. It is claimed to be user expandable for foreign device support. Also, all the UCB's for all error logging devices have been extended to keep on-line error information and a new utility, ELI, is provided to get on-line error information. As mentioned above, a new directive allows any task to log a message to the error log. Also, the I/O exerciser (IOX) has been

improved and replaces user mode diagnostics.

11.0 MISCELLANEOUS FEATURES

* Two 64KW standalone systems are provided for disk backups. DSC64K provides a system with DSC, BAD, FMT, and CNF (autoconfigure). BRU64K is a system with BRU, BAD, and CNF. For RSX-11M Plus only, BRUSYS is a standalone system with everything (DSC, BRU, BAD, FMT, and CNF).

* The indirect command file processor (IND) has many new special symbols. Also, the previously undocumented IND commands are now supported and documented. The version on RSX-11M Plus is now unprivileged.

* Much more help text is available. In addition, a subroutine is provided that allows user programs to access text in the help file.

* The catch-all feature (...CA.) is still available. An unsupported task named TDX is provided. TDX will scan the system UIC for a file name that matches any command it receives and run that task.

* PIP enhancements include wild-character names and file selection by creation date!

* The macro assembler is rewitten and is much faster.

* FTB now supports shared commons and librarys, like FCSRES. This makes FTB usable in most environments and will greatly speed task builds.

* TKB now supports a concept called "cluster libraries". These are resident libraries which share the same mapping register and are mapped on the fly. Cluster libraries may even call each other. For a resident library to be used as a cluster library, it cannot pass or receive any arguments on the stack. The major purpose of cluster libraries is so things like RMS, FCS, and FMS can share the same addresses.

* More crash devices are supported and ^S, ^Q work in the crash routine and XDT.

* The IP11 driver is now loadable.

12.0 RSX-11M PLUS ONLY ENHANCEMENTS

The following list are new features of RSX-11M Plus V2.0 only and do not apply to RSX-11M. Almost all of the above lists applies to both systems, except where noted.

* Some optional features are now standard: secondary pool, parent-offspring support, PLAS directives, multi-user support, and executive level dispatching.

* The task builder and executive now support I/D space for user tasks! This means RSX-11M Plus users can have 64KW tasks (32KW instructions, 32KW data).

* Supervisor-mode libraries map to the user-mode data area. This means RSX-11M Plus users, with a little work, can have 96KW tasks (64KW instructions, 32KW data)!

* Besides the previous support for seek-ahead, the disk drivers now support seek optimization. Three algorithms are available: one-direction only (ramp), one-direction and back (sawtooth), and shortest seek.

* I/O throughput is improved and more tuning is allowed without the need for a system generation.

* Command-level typeahead is supported. Also, the size of typeahead buffers is settable on a per-line basis.

* RSX-11M Plus V2.0 will now support up to 256 terminals.

* Accounting has been improved to add task CPU time limits, memory accounting, and disk performance data. Also, the accounting data is readable by Datatrieve.

* Mixed MASSBUS devices are supported. This means you can have different model disk drives on the same controller.

* Batch supports all DCL commands.

* A new directive is added for variable-length sends with request or connect.

* SAV on RSX-11M Plus can now save all of memory.

* The virtual terminal feature is more terminal-like.

Note, while the PDP-11/24 is supported by RSX-11M Plus V2.0, this does not include any I/D or supervisor-mode features and the number of terminals is limited to 64. Otherwise, full support is provided.

13.0 RSX-11S IMPROVEMENTS

To start with, the version number has been changed to RSX-11S V4.0 to get in step with RSX-11M. Other features include system-controlled partitions, full PLAS support, full 22-bit processor support, and RX02 support in SIP. Also, the system generation supports a target UIC.

14.0 QUESTION AND ANSWER SESSION

Digital made the annoucements at the first session of the symposium. Naturally, there were many questions people had after the annoucements. At one session, a direct phone hookup was made to the development group in Spitbrook and all developers were available to answer the questions. The following compiled list of questions and answers is taken from the rough notes I have of the sessions.

* Is KED to be unbundled from FMS for RSX? KED is and will be a separate product. In addition, the keypad portion of EDT V2.0 is effectively KED.

* Will DSR (Digital Standard Runoff) be distributed for RSX? I do not know. DSR is written in Bliss and is quite large.

* Why are the improvements in the full-duplex terminal driver not also made to the half-duplex version? Many improvements are made to both. However, the full-duplex terminal driver has substantally different data structures and it is not always possible to implement some of its features in the half-dupluex terminal driver.

* Will a RK05 distribution kit be available for RSX-11M V4.0? Yes, however it will be two more disks than RSX-11M V3.2.

* What efforts have been done to improve the documentation for the new releases? Three to four manuals have been modified or rewritten (MCR, DCL, Exceutive Reference), but much work will be done in the future.

* What executive data structures change in RSX-11M V4.0? The UCB for error logging devices has addition words. Also TCB's and task headers are modified to support asynchronous buffered I/O and full event flag support.

* Why is the new autobaud detection limited to 1200 baud? The 1200 baud limit is what can be reliably done. Certainly higher baud rate detection is possible but it cannot be done reliably yet.

* Will there be a symbolic debugger for the new releases? No.

* Can a RSX-11M V4.0 system generation be done on-line with our RSX-11M V3.2 systems? Not totally yet, but the problems are minor and should be fixed by the release date.

* In PIP's selective delete mode, in which order will the files be displayed? In the order they appear in a PIP directory.

* Will the VAX application migration executive (AME) be updated for RSX-11M V4.0? We are trying hard for this.

* Will DECnet Phase III be supported on RSX-11M V4.0? Yes, however a patch file for DECnet will be distributed that must be applied first.

* Has any task builder changes been made in RSX-11M V4.0. No major changes, just some new switches and cluster library support.

* Will there be a system logic manual for RSX-11M V4.0. Yes, but not in the initial distribution. The manual may be distributed in pieces so we can get the material out as fast as possible.

* Will there be a price break for the RSX-11M Plus single RL02 distribution kit? Yes, but size is unknown.

* Can BRU64K or DSC64K restore backups from RSX-11M V3.2 to disk of different sizes? Yes, if size of output disk is larger or equal to original disk.

* Will QMG support despooling from magtape? Do not know.

* How much longer will a RSX-11M V4.0 system generation take? The best guess is about 30 minutes to an hour longer. The extra time is needed for generating the all the privileged and utility task build command files.

* Up to now, we have relied on the fact that a running system would fit on a single RK05. Will this be true for RSX-11M V4.0? Yes, if the same functionality is selected. It is unclear how much disk space the new error logging facility takes.

* Will there be MCR commands to change the round-robin and disk swapping parameters? Not in RSX-11M V4.0, however, available in RSX-11M Plus V2.0.

* Could you explain the policy used to select what sources to put on the source kits? Things like EDT, RMS, and RMS-11K are not available. The reason for these ommissions is that languages not normally available are used (Bliss for EDT, SUPMAC for RMS).

* Can DCL be a catch-all task for MCR? Also can MCR be a catch-all task for DCL? The catch-all task mechanism will be documented, but using DCL or MCR as catch-all's is not supported.

* With all the user interest in writing ACP's, is the QIO/ACP interface documented? Yes, documentation will be provided for both general user-written ACP's and the Files-11 QIO's. [ED: Free at last, free at last]

* Are the DV.SPO/DV.SPI (output spooled device/input spooled device) in U.CW1 used by anything? Yes in RSX-11M Plus, but not currently in RSX-11M. However, a future release may use them.

* Have there been any changes in the KLAB and LPS libraries for RSX-11M V4.0.? No.

* Will a future release of RSX-11M supported overlapped seeks or will this remain a privilege reserved for RSX-11M Plus users? Overlapped seeks require data structures which exist onlt in RSX-11M Plus. Adding these data bases to RSX-11M would tax the pool too heavily. The

feature must therefore remain restricted to RSX-11M Plus.

* Now that 6250 BPI tape drives are available for the VAX, are there any plans to include support for them in RSX-11M? No.

* Will the modification to REQSB in the August 1981 Software Dispatch to improve partition allocation algorithm (ignore task with I/O in pass 1) be included in RSX-11M V4.0? Yes.

* With the current BRU, the directory listings only go to the terminal. Has this been changed so they can be spooled or output to a file? No, maybe in a future release.

* Will the multi-user switch in TKB for RSX-11M Plus be supported in RSX-11M V4.0? No.

* Could you document how to use FCS big buffering in a Fortran environment? There has been problems in RSX-11M V3.2 with the .POINT routine that affect big buffering. These are fixed in RSX-11M V4.0, however, there still will be no documention.

* Where will RSX-11M V4.0 keep its help files? On LB:[1,2].

PRESERVE, DSC, BRU

Dale Donchien

Software Developer
Digital Equipment Corporation

1.0 PRESERVE

The Volume Preservation Utility (PRESERVE) is useful for copying a volume written in any format. It supports magnetic tape, DECtapes, cassette tapes, and most but not all disks. Copies can be made in any of three formats: Files-11, logical tape, and image mode. Image mode copies all the blocks from the input device to the output device and can be used to save data regardless of its organization. Files-11 mode can only be used with disks and DECtapes which are valid Files-11 volumes. In this mode only the allocated data blocks are transfered to the output device. Logical tape mode writes a hardware bootable copy of PRESERVE on the output device in addition to the data. This is the default mode when copying from a disk or DECtape to a magnetic tape.

PRESERVE is a stand-alone system which requires that the online system be shutdown before PRESERVE is used. Various switches are provided to establish

the CSR and vector of the supported devices. Other switches can be used to display help text and to request a verification pass to insure the copy was successful.

PRESERVE has many shortcomings. It does not recognize bad disk blocks in image mode, thus it will attempt to copy them as if they were data. In Files-11 mode, it copies each allocated block from the input device to the same logical block on the output device, therby not compressing the output copy (contiguous space is not reclaimed). Also, problems result when the output device has a different block capacity than the input device. For example, if a small disk is copied to a large disk, the additional blocks on the large disk will be wasted.

With the advent of DSC and BRU, there are few reasons to use PRESERVE to copy a Files-11 volume. It is, however, very useful when an exact duplicate of a volume is required, or when a non-Files-11 volume needs to be copied.

2.0 DSC

DSC, the Disk Save and Compress utility, is an improvement to its predecessor, PRESERVE, in that it handles bad blocks, supports all disk and tape drives, and runs both stand-alone and online. Like PRESERVE, it does full disk backups, but DSC cannot be used for volumes which are not written in Files-11 format. DSC also compress the output volume by copying the files which are scattered over the input disk to contiguous areas on the output volume. DSC will copy a disk to a tape, from tape to another disk, from a disk to another disk directly, and from a DSC format tape to another tape. As with PRESERVE, a verification pass is optional and switches are provided in the stand-alone version to change the device configuration. DSC also featues a compare capability to verify that two volumes are alike without doing any data transfer.

DSC is driven by the index file ([0,0]INDEXF.SYS). It locates the index file on the input disk, and then finds the files one by one from their headers. For each file, the header is copied, immediately followed by its data, so files are contiguous on tape and the output disk. File directories are copied as is. They are not treated specially by DSC nor do they need to be since they look like just another file. As a result, DSC does copy lost files (files not in any directory), however, they remain lost when copied to the output volume. The output disk is pretty much an exact copy of the input disk, except that it is compressed. Characteristics of the input volume such as the maximum number of files and volume label become the same for the output disk and cannot be changed. DSC does perform validity checks on the file headers to make sure they are valid.

DSC uses double buffering for increased through-put, with a default buffer size of 2K bytes (4 disk blocks). A block factor switch is provided which can be used to increase the number of buffers used, up to a maximum determined by the installed size of the task image. The extra buffers serve to increase performance, however the large number of disk seeks necessary to copy the input disk slow down the backup process, especially on disks which have few contiguously allocated files.

3.0 BRU

The Backup and Restore utility, BRU, is the succesor to DSC. It is optimized for speed at the sake of memory. It contains a superset of DSC functionality, plus it implements many additional features including backup and restore from/to a mounted disk and incremental backup and restore. Like DSC, it can only be used to copy Files-11 volumes and during the copy, the output disk will be compressed. Unlike both PRESERVE and DSC, due to its large memory requirements there is no stand-alone version of BRU for RSX-11M. A BRU with subset functionality is available with RSX-11M Plus in a stand-alone system which includes the formatter (FMT) and bad block locator (BAD) utilities, but this only runs on a processor with memory management support and 96K words of memory.

Unlike DSC, BRU is directory driven to allow for incremental operation. Incremental processing permits only a specified category of files to be copied, rather than the whole disk. Up to 16 file specifications can be entered to select files which should be transferred. BRU locates the files by finding their entries in the directory, which points to the file header in the index file. As with DSC, BRU checks the file header for validity. Since BRU will only find files which are contained within a directory, it will not find lost files and thus they will not be copied. It is recommended that the VFY utility be invoked to find any lost files and enter them in a directory before BRU is used to copy the disk, otherwise the lost files will be lost for good. The directories themselves are not copied to the output disk. BRU constructs them as the files within them are transferred.

BRU uses double buffering for speed in the same way that DSC does, however, the size of the buffers are twice as large but cannot be larger as with DSC. The speed improvement which BRU realizes is not a function of buffer size as much as its use of a unique algorithm for determining the order in which the data blocks should be read from the input disk. This algorithm, explained below, keeps the number of disk seeks to a minumum by reading the data in increasing logical block order.

When copying the data from the input disk, BRU uses an internal table which is filled in from the file headers of the files being copied. BRU reads the file headers into its buffers, and for each header it saves the retrieval pointers and the file-ID. The retrieval pointers are that part of the header which identifies where, by logical block number, the file resides. The file-ID is a number which indentifies the file header, and thus the file itself. As each header is read and the retrieval pointets extracted, the header is written to the output volume. When the table is full, it is sorted by logical block number. Then the data is read according to the sorted logical block information in the table. Since the reads are issued in increasing block order, the disk head moves linearly into the disk without performance loss due to erratic disk head movement.

During its restore phase, when the output disk is written, the same table is built from the retieval pointers of the input disk. At the same time, space is allocated for the file on the output disk, as contiguously as possible. The retrieval pointers thus formed are inserted into the table and are also copied into the file header on the output disk. When the table is full, it is sorted by the input disk retrieval pointers, which results in the order the data was read from the input disk. The data is read and written to the output disk

according to the output disk retrieval pointers in the table. Since space for the output files are allocated contiguously, the output disk is compressed. The directories are created on the output disk as each new directory is encountered on the input volume. An entry in the directory is made each time a file header is processed. Note, that with this scheme, file ehaders and their data are not together on the tape, and all of a file's data is also not necessarily together on the tape.

Unlike PRESERVE and DSC, BRU cannot perform tape to tape copies, however, PRESERVE will copy a BRU format tape using image mode. Also unlike both PRESERVE and DSC, BRU allows disk characteristics to be changed when the output volume is initialized. Parameters such as the maximum number of files, volume label, and position of the index file can be specified. It is necessary to change the parameters of the output disk when copying to a different disk type since by default BRU will initialize the output disk with the parameters of the input disk, although they are not the same for different disk types.

BRU and DSC can both compare two volumes to see if thet are alike without performing any data transfer. DSC does not efficiently handle multi-header files where the number of file headers is different on the input disk than on the output disk. The file will be copied correctly but with some wasted space. BRU can dynamically change the number of file headers used by multi-header files. It will delete extra headers if not needed and add extra headers if required and spare headers exist. DSC will not allocate space to a contiguous file if the space in not available. BRU will allocate it in non-contiguous blocks, restore the file, and issue a warning.

BRU is the only backup utility of the three which allows the backup and restore of a disk which is mounted and in use by the system. Since files can be created, deleted, and modified on a mounted disk while BRU is running, there is no guarantee that the BRU copy accurately reflects the input disk. Since BRU issues logical I/O requests to read the input disk, BRU must be built with the privilege attribute to be able to copy from a mounted disk. During restore to a mounted disk, BRU issues file level directives to create files, thus there should be no problem with contention for the disk by other tasks.

4.0 CONCLUSION

PRESERVE has been outdated by the availibility of disk drives it does not support and newer backup utilities which offer greater speed and features. It's only unique use is for image mode copies of volumes which are not in Files-11 format, such as duplicating BRU format tapes. DSC is necessary for systems which do not have memory management and for stabd-alone use. BRU offers the most number of features but it is large. For RSX-11M Plus systems, BRU is always recommended.

F11ACP Performance Measurements for RSX-11M

Joseph S. Sventek
Computer Science & Mathematics Department
Lawrence Berkeley Laboratory
Berkeley, CA 94720

One of the many questions presented to an RSX-11M system generator during Phase II of SYSGEN is:

```
>* 11. FCP desired (Type <ESC> for explanation) [D: MIDDLE] [S]:<ESC>
>;
>;    Five FCP's are supported by SYSGEN.  They are:
>;
>;    Non-multiheader  - Minimum functionality (Size: 2K)
>;    Small            - Full functionality, heavily overlayed (2.25K)
>;    Middle           - Default, optimized for RMS, open/close envir. (5.2K)
>;    Large            - Optimized for all applications (6.7K)
>;    PLAS             - Fastest possible, optimized for create/delete (10.1K)
>;
```

Little in the way of direct information concerning performance vs. memory occupancy is provided, other than the terse one-liners above. Thus, one is forced to rely upon the intuition that the larger ACP's should perform better, since less time is spent bringing in the appropriate overlay. As a result, the system generator usually opts for as large an ACP as the system can tolerate in hopes of maximizing system performance.

In order to truly "maximize system performance", a metric is needed to permit comparison of the performance of systems which are identical except for the inclusion of different ACP options. Due to the vastly different hardware configurations which populate the RSX-11M community, it is difficult to devise metrics which give accurate absolute measures; even so, one should be able to provide quantitative relative measures on a particular machine, and thereby aid the system generator in the choice of the option most appropriate to the particular system.

The metric for ACP effectiveness used in this paper is based upon the following two principles:

1. The effectiveness of an ACP is large if the time necessary to perform a given benchmark is small. Symbolically,

   effectiveness ~ 1 / time

2. The ACP effectiveness is large if the size of the GEN partition is large. Symbolically,

   effectiveness ~ size(GEN)

   where size(GEN) = total memory - (size(RSX_EXEC) + size(ACP)).

Combining these two relationships, the effectiveness metric is defined as

effectiveness ~ size(GEN) / time

In the limit where the total memory of the system is large (eg. 1 MB 11/70), the ACP effectiveness should be dominated by the 1/time term, since different sized ACP's will have little effect upon size of the GEN partition.

It is difficult to determine the constant of proportionality to arrive at an absolute effectiveness, but taking the ratio of one ACP's effectiveness to some standard ACP's effectiveness permits the determination of the relative effectiveness. Since the MIDDLE ACP is the default ACP for mapped systems, all effectivenesses listed in the tables and plotted in the figures are relative to the effectiveness of a system with a single copy of the MIDDLE ACP.

This metric has been used to compare several RSX-11M v.3.2 systems with different ACP options included. The hardware configuration for the tests was an 11/34A CPU, 124KW of core memory, and an AED controller driving two AMPEX DM980 80-MB storage modules. The AED controller makes each storage module look like a 67MB RP03. Two benchmark scripts were run on the otherwise quiescent systems:

1. DCHECK.CMD - Simultaneous wildcard PIP directory listings of the two disks are performed. This simply tests the directory access portion of the ACP's.

2. DWRITE.CMD - Several different PIP commands are performed simultaneously on the two disks, which is meant to exercise most of the overlays of the ACP's.

Appendix A may be consulted for the exact contents of these command files. In all cases, DP0: was the system disk, with 4490 files in 180 UIC's occupying 127000 blocks; DP1: was identical to DP0:.

In order to calculate the effectiveness of the ACP's, it is necessary to calculate the size of the resulting GEN partition. For the measurements described in this paper, the following partitions made up the RSX Executive:

EXEC	120000
LDRPAR	2500
DRVPAR	2000
TTPAR	20000
SYSPAR	10000

This corresponds to a 20K executive, one disk driver, a 4KW full-duplex terminal driver, and space for MCR... and the task loader. The system is then completed by inclusion of the particular ACP option, leaving the remainder of memory to the GEN partition.

Four ACP's were tested in various configurations:

1. MD1 - this is the ACP one gets when MIDDLE is specified to the SYSGEN question quoted above. This is usually the default ACP for mapped systems.

2. LG1 - the ACP one gets when the answer to question 11 is LARGE.

3. RO2 - the ACP one gets when the answer to question 11 is PLAS.

4. CO1 - this is a core-only ACP which was derived from [1,24]fcpco0bld.cmd and [1,24]fcpcorbld.odl. The files [1,24]fcpco1bld.cmd and [1,24]fcpco1bld.odl are provided in appendix B.

Tables I and II present the data for the two benchmarks executing on systems with various ACP options. (Figures I and II are pictorial representations of the tables.) The first four rows of the tables present the data for systems which contain a single copy of the relevant ACP. Two relative effectivenesses are displayed: one for a 124KW 11/34, and a second assuming 512KW of memory, as in a 1MB 11/70. The total memory size affects the size of the GEN partition, and the "11/70" column is included to show that the relative effectiveness is dominated by the 1/time term.

Two interesting trends can be observed from the data:

1. For DWRITE, which is the more realistic of the two scripts, it is the case that the order of effectiveness is MD1 < LG1 < RO2. On an 18-bit machine, CO1 is actually more effective than RO2, while the opposite is true on a 22-bit machine.

2. For DCHECK, the resident overlaid ACP is actually slower than the disk overlaid ones, since it must dynamically map itself to its data buffers.

There have long been rumors that RSX-11M systems perform better when each disk drive has a dedicated ACP. When a single ACP is servicing multiple disks, much of the directory context of one disk is lost if the ACP needs to service one of the other disks, with the result that when attention is returned to the original disk, that context must be re-established via extra disk activity. By dedicating an ACP/disk, this loss of context can be avoided.

To test this hypothesis, four more systems were built, each with an ACP per disk drive, using MD1, LG1, RO2 and CO1, respectively. The second four rows of tables I and II show the results of the measurements on these systems. The message from these measurements is clear: it is much more effective to dedicate an ACP per disk, even on 18-bit machines. On 18-bit machines, 2 copies of CO1 is more effective than RO2, while the

reverse is true for 22-bit machines. In the case of DCHECK, it is still the case that the resident overlaid system is slower than the others.

Even though this metric indicates that it is more effective to use the extra memory for additional ACP's (due to the increased speed of execution), owners of 18-bit machines may still balk at dedicating ~30000 bytes of memory for the file system, since the GEN partition can become intolerably small. The only way in which multiple ACP's may appeal to these owners is if the memory cost/ACP becomes small. This can be achieved by building a multi-user F11ACP.

A multi-user task is one in which the pure code and pure data have been segregated from the impure code and data; the resulting read-only segment is mapped by a separate APR. In this way, many copies of the read-write portion of the task can be simultaneously active, mapped over a single copy of the read-only segment. If the size of the read-only segment is represented by RO and that of the read-write is RW, then the memory savings by building the task multi-user is

$$n * (RW + RO) - (RO + n * RW) = (n - 1) * RO$$

where n is the number of copies of the task that are simultaneously active.

An article by Eric Levy of JPL in the February 1981 Multi-tasker described the steps necessary to build a multi-user task for use under RSX-11M. The most recent distribution of the Software Tools Virtual Operating System (Spring 1981 RSX Symposium Tape) provides an utility (MUBLD) which performs the operations described in the article. To build a multi-user version of F11ACP, one must:

1. Patch TKB as per the June 1981 Software Dispatch for RSX-11M+ on pages 123-126. These patches permit TKB to correctly build a priveleged multi-user task. Note that the correct checksums for the ".POB" modules are

   P2POP.POB 21461
   SGALO.POB 22734

2. Assemble FCPMU.MAC provided in appendix C below. These PSECT declarations (which were provided by Larry Baker of USGS, Menlo Park, CA) serve to define the read-only psects.

3. TKB @FCPMUBLD, using the patched version of TKB and the files FCPMUBLD.CMD and FCPMUBLD.ODL provided in appendix C below. This results in a task image FCPMU.TSK which has the RO code collected together and mapped by APR6. This task image is input for MUBLD. (The warning messages from TKB concerning multiply defined PSECTS can be safely ignored.)

4. If it has not yet been done, build the software tools system from the Spring 1981 RSX SIG Tape. After the

TOOLGEN is completed, it is nessary to fix a bug in MUBLD with the following sequence of commands:

```
>"Get a copy of [307,33]mubld.w into the current directory"
>SHL
% ar xv mubld.w
% ar xv mubld.r domu
% ed domu
: /PIDSIZE/
: s//20/p
: w
: q
% ar uv mubld.r domu
% rc -v mubld.r
% logout
>
```

5. >SHL -C MUBLD -V FCP
   Mubld extracts the RO code into a resident-library style task image named rofcp.tsk. It also generates another task image mufcp.tsk which has the RW code and is mapped over a common partition named ROFCP for its RO code.

6. Look at fcpmu.map for the sizes of the two sections of code:

   ```
   R/W MEM LIMITS:  120000  135623  015624  07060.
   R-O MEM LIMITS:  140000  157277  017300  07872.
   ```

   (These are the numbers one should see if the EXTSCT's in fcpmubld.cmd are as they appear in appendix C. As such the size of the RO segment is 017300 and that of the RW is 015700.)

7. Armed with the sizes of the two code sections, VMR a new system image with the appropriately sized partitions. For example, if two ACP's are desired, with the device name of the second disk being DB1:, then the appropriate VMR commands are:

   ```
   SET /MAIN=ROFCP:*:173:COM
   INS ROFCP
   SET /MAIN=FCPPAR:*:336:SYS        ! 33600 = 2 * 15700
   INS MUFCP/TASK=F11ACP/CKP=NO
   INS MUFCP/TASK=DB01F1/CKP=NO
   ```

   The copies of mufcp must go into a system partition, as tasks in a TASK partition cannot be mapped over a common region.

8. After the initial boot of the system, execute

   ```
   >FIX F11ACP
   >FIX DB01F1
   ```

   before saving the system. VMR does not permit fixing of

tasks in SYS partitions.

9. The normal mount of DB1: in startup.cmd will cause DB01F1 to be used, since the order of ACP precedence when mounting a disk is:

   i. If /ACP=XXXXXX in MOU command, use XXXXXX
   ii. If this is ddnn: and DDNNF1 is installed, use DDNNF1
   iii. Use F11ACP

A system with two copies of the multi-user ACP was generated, and the same two benchmarks were run. The results of these benchmarks can be found in the ninth row of tables I and II. Since MUFCP is simply a multi-user version of FCPC01, it is not surprising that the execution times are identical with the system with two copies of C01. It is also apparent that the MUFCP system is the most effective ACP option in nearly every situation. The author's standard system is now the multi-user system, and has been running for more than one month without any problems.

In conclusion, it is true that in multiple disk systems (program development engines), an ACP/disk provides a large performance enhancement over a single, large ACP, whether disk or memory overlaid. On memory-tight systems, this can be achieved by building a multi-user version of F11ACP.

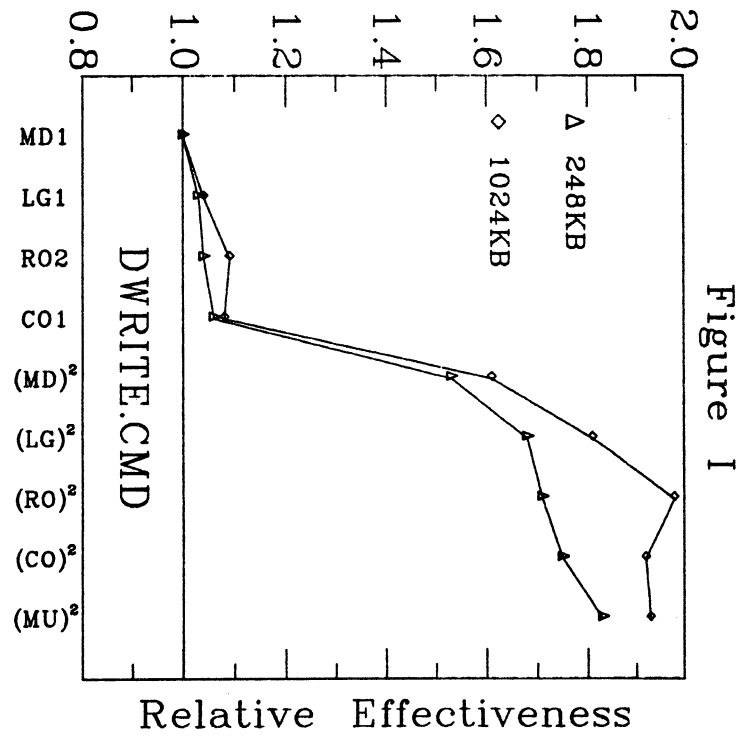

Figure I

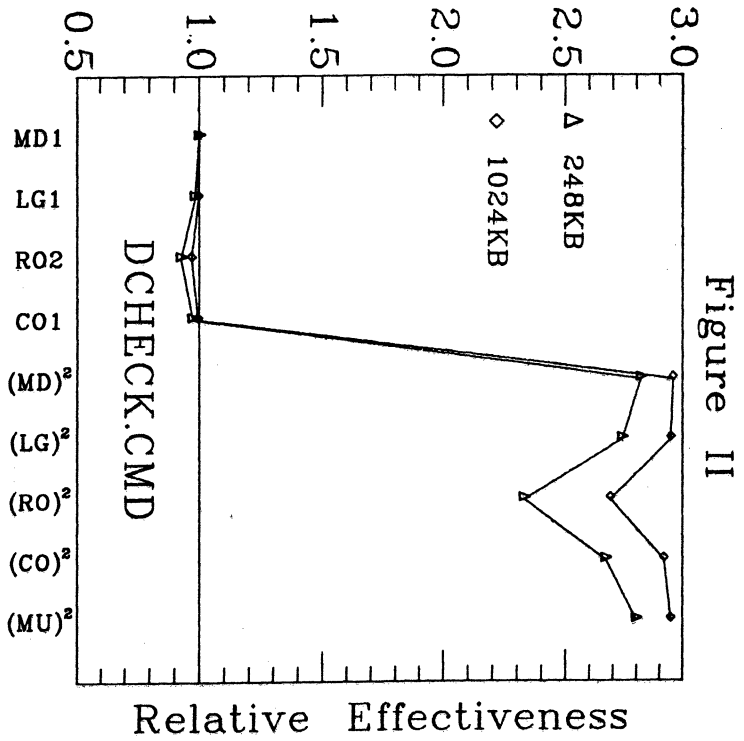

Figure II

Table I

ACP Effectiveness for DWRITE.CMD Benchmark

System	Size(bytes)	Time(secs)	11/34 eff.	11/70 eff.
MD1	10304	651	1.000	1.000
LG1	13312	624	1.026	1.040
RO2	20480	591	1.042	1.090
CO1	14912	599	1.060	1.082
2 * MD1	20608	401	1.534	1.606
2 * LG1	26624	354	1.679	1.808
2 * RO2	40960	318	1.713	1.982
2 * CO1	29824	333	1.752	1.915
2 * MU1	22080	334	1.827	1.925

Table II

ACP Effectiveness for DCHECK.CMD Benchmark

System	Size(bytes)	Time(secs)	11/34 eff.	11/70 eff.
MD1	10304	971	1.000	1.000
LG1	13312	971	0.984	0.997
RO2	20480	991	0.927	0.969
CO1	14912	971	0.975	0.995
2 * MD1	20608	325	2.824	2.956
2 * LG1	26624	324	2.737	2.946
2 * RO2	40960	349	2.328	2.693
2 * CO1	29824	326	2.669	2.918
2 * MU1	22080	325	2.800	2.951

Appendix A - Benchmark Command Files

DWRITE.CMD

```
INSTALL $PIP/TASK=...POP
INSTALL $PIP/TASK=...P1P
TIME
.XQT POP @DP0:[2,154]POP
.XQT P1P @DP1:[2,154]P1P
.WAIT ...POP
.WAIT ...P1P
TIME
REMOVE ...POP
REMOVE ...P1P
```

DCHECK.CMD

```
INSTALL $PIP/TASK=...POP
INSTALL $PIP/TASK=...P1P
TIME
.XQT POP NL:=DP0:[*,*]/LI
.XQT P1P NL:=DP1:[*,*]/LI
.WAIT ...POP
.WAIT ...P1P
TIME
REMOVE ...POP
REMOVE ...P1P
```

DP0:[2,154]POP.CMD

```
DP0:[2,377]=DP0:[307,31]*.*;*
NL:=DP0:[2,377]/FU
NL:=DP0:[2,377]*.CMD
DP0:[2,377]*.*/PU
DP0:[2,377]*.*;1=DP0:[2,377]*.*;*/RE
DP0:[2,377]*.*;*/DE
```

DP1:[2,154]P1P.CMD

```
DP1:[2,377]=DP1:[307,31]*.*;*
NL:=DP1:[2,377]/FU
NL:=DP1:[2,377]*.CMD
DP1:[2,377]*.*/PU
DP1:[2,377]*.*;1=DP1:[2,377]*.*;*/RE
DP1:[2,377]*.*;*/DE
```


Appendix B - Files for Building FCPC01

FCPC01BLD.CMD

```
;       F C P C 0 1 B L D . C M D
;
; BUILD CORE RESIDENT F11ACP (FCPC01) FOR MAPPED RSX11M
;       NAME - FCPC01
;       .ODL - FCPCORBLD.ODL
;       SIZE - 6. KW
;       FCB LIST - 32 FCB'S
;       SEPARATE DIRECTORY BUFFERS - 3
;
;
TK:[1,54]FCPC01/AL/AC/MM/-FP,MP:[1,34]FCPC01/-SP=FCPC01BLD.ODL/MP
;
; OPTIONS INPUT
;
TASK=F11ACP
STACK=32
UNITS=1
UIC=[1,1]
PRI=149
PAR=FCPPAR:0:0
;       PAR=FCPPAR:0:30100
TSKV=.SSTVC:7
;
; THE MAXIMUM SIZE FOR THIS FCP IS 12K. ONLY THE DIRECTORY BUFFER, HOWEVER,
; CAN LIE IN THE LAST 4K (APR7).  THE FOLLOWING BUFFERS MUST BE IN THE LOWER 8K
; (APR5 AND APR6).
;
;
; ALLOCATE A SEPARATE BUFFER FOR INDEX FILE BITMAP
;
;       EXTSCT=$$BUF0:1006
;
; DEFAULT IS ONE BUFFER SHARED BETWEEN BITMAP AND FILE HEADERS (AND DIRECTORY
; BLOCKS).  THIS ACTION WILL SPEED UP APPLICATIONS THAT DO A LOT OF FILE
; CREATION AND DELETION.
;
;
;
; ALLOCATE A SEPARATE BUFFER FOR FILE HEADERS
;
EXTSCT=$$BUF1:1006
;
; DEFAULT FROM ASSEMBLY IS ONE BUFFER SHARED BETWEEN FILE HEADERS
; AND DIRECTORY BLOCKS (AND INDEX FILE BITMAP).  THIS ACTION WILL BENEFIT ANY
; APPLICATION THAT CAN AFFORD THE MEMORY.
;
;
;
; ALLOCATE PRIVATE FCB POOL SPACE
;
; EACH FCB = 52 OCTAL
```

```
;
;       EXTSCT=$$AFR1:XXXX
;
;       ,WHERE XXXX = N*52 (OCTAL) AND N = NUMBER FCB'S DESIRED
;
EXTSCT=$$AFR1:2500
;
;
; *** NOTE ***
; THE PRECEDING BUFFERS AND EXTSCT'S MUST LIE IN ADDRESS SPACE BELOW APR7.
; DO NOT EXTSCT PAST 157777 OCTAL!!
;
; THE DIRECTORY BUFFER BELOW MAY LIE, IN WHOLE OR IN PART, WITHIN THE ADDRESS
; SPACE OF APR7.  CONSEQUENTLY, THE MAXIMUM ALLOWABLE FOR THIS EXTSCT IS
; <3.5K + N> OR <17000 (OCTAL) + N> WHERE N REPRESENTS THE AMOUNT OF SPACE UNDER
; APR7 THAT IS STILL AVAILABLE AFTER THE PRECEDING EXTSCT'S HAVE BEEN APPLIED.
;
;
; ALLOCATE ADDITIONAL BUFFER SPACE FOR DIRECTORY BLOCKS
;
;       EXTSCT=$$BUF3:XXXX
;
;       ,WHERE XXXX REPRESENTS 1000 FOR EACH 'ADDITIONAL' BLOCK
;       DESIRED.
;
; DEFAULT FROM ASSEMBLY IS ONE DIRECTORY BLOCK.  NOTE THAT THIS
; ACTION SHOULD BE IN ADDITION TO ALLOCATING A SEPARATE BUFFER
; FOR FILE HEADERS.  ALLOCATING ADDITIONAL BUFFER SPACE FOR DIRECTORY
; BLOCKS WITHOUT A SEPARATE BUFFER FOR FILE HEADERS DOES NOT
; ACCOMPLISH VERY MUCH.
;
EXTSCT=$$BUF3:2000
;
;
;
/
```

FCPCO1BLD.ODL

```
;       F C P C O R B L D . O D L
;
;       OVERLAY DESCRIPTION FOR CORE RESIDENT F11ACP - RSX11M/M+
;
;               MINIMUM SIZE -- 6.0/6.25 K  PHYSICAL
;               MAXIMUM SIZE -- 12 K  PHYSICAL
;
;       THIS ODL BUILDS A CORE RESIDENT F11ACP.  IT DOES NOT REQUIRE PLAS.
;       IT IS INTENDED FOR USE IN SYSTEMS WITHOUT PLAS OR WITH APPLICATIONS
;       THAT WANT A SMALL RESIDENT F11ACP (THIS ODL DOES NOT REQUIRE
;       A SEPARATE BUFFER FOR FILE HEADERS LIKE RESIDENT OVERLAY ODL'S).
;       NOTE THAT IT PROVIDES A MINIMUM OF SPACE AVAILABLE FOR FCB'S.
;
```



```
;
;       STRUCTURE:
;               APR5 - ROOT - CODE
;               APR6 - ROOT - CODE
;               APR6 - CO-TREE - BUFFERS
;               APR7 - CO-TREE - DIRECTORY BUFFER  (MAY OVERMAP I/O PAGE)
;
;
        .NAME   F11ACP
        .ROOT   F11ACP-FCPROT-FCPCO
;
;       THE FOLLOWING MODULES MUST BE IN THE ROOT
FCPROT: .FCTR   FCP/LB:F11ACP:F11CM:SMCOM:DISPAT-FCPRT2
FCPRT2: .FCTR   FCP/LB:ARWVB:OVERR:FCPGBL-CODE
;
;       CO-TREE CONTAINING DIRECTORY BUFFER OVERMAPS APR7
FCPCO:  .FCTR   FCP/LB:F11BUF:DIRBUF:INIT
;
;       THESE ARE THE REST OF THE MODULES FOR FCP
CODE:   .FCTR   FCP/LB:ALLOC:BLXIO:CKSUM:CLACC:CLDAC:CLDIR:CRFID-CODXX1
CODXX1: .FCTR   FCP/LB:DARITH:DLMRK:DRACC:DREOF:DREX:DRGET:DRINI-CODXX2
CODXX2: .FCTR   FCP/LB:DRPAC:DRSEF:DRVLB:DRWRT:DWPND:FDRMV:FNDNM-CODXX3
CODXX3: .FCTR   FCP/LB:GTFID:GTMAP:INFCB:INWIN:LOCAT:MPHDR:MPVBN-CODXX4
CODXX4: .FCTR   FCP/LB:NXHDR:PROCK:RDHDR:RLEAS:RLFCB:RWVB:RWVBL:RW1LB-CODXX5
CODXX5: .FCTR   FCP/LB:SCFAC:SCFCB:SMALC:SMNXB:SMRVB:SMSCN:WACCK:WITRN-CODXX6
CODXX6: .FCTR   FCP/LB:WRHDR:WTRN1:ACCESS:ATCTL:CRFIL:DATIM:DEACC:ENTNM-CODXX7
CODXX7: .FCTR   FCP/LB:EXCMP:EXCOM:EXTEN:RATCM:RDATT:RWATT:WATCM:WRATT-CODXX8
CODXX8: .FCTR   FCP/LB:CLDEL:CLATT:CLCRE:CLEXT:CLFCB:CLNUP:DLBLK:DLFIL-CODXX9
CODXX9: .FCTR   FCP/LB:DLHDR:DRALC:DRCPY:DREXT:EXTHD:IXEXT:RMVNM:SMDEL-CODX10
CODX10: .FCTR   FCP/LB:TRUNC:DMOUNT-[1,1]EXELIB/LB-[1,54]RSX11M.STB/SS
;
        .END
```

Appendix C - Files for Building Multi-user FllACP

FCPMU.MAC

```
        .title  fcpmu
        .ident  /1.0/

        .psect  access  ro,i,lcl,rel,con
        .psect  alloc   ro,i,lcl,rel,con
        .psect  atctl   ro,i,lcl,rel,con
        .psect  blxio   ro,i,lcl,rel,con
        .psect  cksum   ro,i,lcl,rel,con
        .psect  clatt   ro,i,lcl,rel,con
        .psect  clcre   ro,i,lcl,rel,con
        .psect  cldac   ro,i,lcl,rel,con
        .psect  cldel   ro,i,lcl,rel,con
        .psect  cldir   ro,i,lcl,rel,con
        .psect  clext   ro,i,lcl,rel,con
        .psect  clfcb   ro,i,lcl,rel,con
        .psect  clnup   ro,i,lcl,rel,con
        .psect  crfid   ro,i,lcl,rel,con
        .psect  crfil   ro,i,lcl,rel,con
        .psect  datim   ro,i,lcl,rel,con
        .psect  deacc   rw,i,lcl,rel,con
        .psect  dispat  rw,i,lcl,rel,con
        .psect  dlblk   ro,i,lcl,rel,con
        .psect  dlfil   ro,i,lcl,rel,con
        .psect  dlhdr   ro,i,lcl,rel,con
        .psect  dlmrk   ro,i,lcl,rel,con
        .psect  dmount  ro,i,lcl,rel,con
        .psect  dracc   ro,i,lcl,rel,con
        .psect  dralc   ro,i,lcl,rel,con
        .psect  drcpy   rw,i,lcl,rel,con
        .psect  dreof   ro,i,lcl,rel,con
        .psect  drex    ro,i,lcl,rel,con
        .psect  drext   rw,i,lcl,rel,con
        .psect  drget   ro,i,lcl,rel,con
        .psect  drini   ro,i,lcl,rel,con
        .psect  drpac   ro,i,lcl,rel,con
        .psect  drsef   ro,i,lcl,rel,con
        .psect  drvlb   ro,i,lcl,rel,con
        .psect  drwrt   ro,i,lcl,rel,con
        .psect  dwpnd   ro,i,lcl,rel,con
        .psect  entnm   ro,i,lcl,rel,con
        .psect  excmp   ro,i,lcl,rel,con
        .psect  excom   ro,i,lcl,rel,con
        .psect  exten   ro,i,lcl,rel,con
        .psect  exthd   ro,i,lcl,rel,con
        .psect  fcpgbl  ro,i,lcl,rel,con
        .psect  fdrmv   ro,i,lcl,rel,con
        .psect  fndnm   ro,i,lcl,rel,con
        .psect  fllacp  ro,i,lcl,rel,con
        .psect  fllcm   rw,i,lcl,rel,con
        .psect  gtfid   ro,i,lcl,rel,con
        .psect  gtmap   ro,i,lcl,rel,con
```



```
        .psect  infcb   ro,i,lcl,rel,con
        .psect  inwin   rw,i,lcl,rel,con
        .psect  ixext   ro,i,lcl,rel,con
        .psect  locat   rw,i,lcl,rel,con
        .psect  mphdr   ro,i,lcl,rel,con
        .psect  mpvbn   ro,i,lcl,rel,con
        .psect  nxhdr   ro,i,lcl,rel,con
        .psect  prock   ro,i,lcl,rel,con
        .psect  ratcm   ro,i,lcl,rel,con
        .psect  rdatt   ro,i,lcl,rel,con
        .psect  rdhdr   ro,i,lcl,rel,con
        .psect  rleas   ro,i,lcl,rel,con
        .psect  rlfcb   ro,i,lcl,rel,con
        .psect  rmvnm   ro,i,lcl,rel,con
        .psect  rwatt   ro,i,lcl,rel,con
        .psect  rwvb    ro,i,lcl,rel,con
        .psect  rwvbl   rw,i,lcl,rel,con
        .psect  rwllb   rw,i,lcl,rel,con
        .psect  scfac   ro,i,lcl,rel,con
        .psect  scfcb   ro,i,lcl,rel,con
        .psect  smalc   ro,i,lcl,rel,con
        .psect  smcom   rw,i,lcl,rel,con
        .psect  smdel   ro,i,lcl,rel,con
        .psect  smnxb   ro,i,lcl,rel,con
        .psect  smrvb   ro,i,lcl,rel,con
        .psect  smscn   ro,i,lcl,rel,con
        .psect  trunc   ro,i,lcl,rel,con
        .psect  wacck   ro,i,lcl,rel,con
        .psect  watcm   ro,i,lcl,rel,con
        .psect  witrn   ro,i,lcl,rel,con
        .psect  wratt   ro,i,lcl,rel,con
        .psect  wrhdr   ro,i,lcl,rel,con
        .psect  wtrnl   rw,i,lcl,rel,con
        .psect  dirbuf  ro,i,lcl,rel,con
        .psect  fllbuf  ro,i,lcl,rel,con
        .psect  init    ro,i,lcl,rel,con
        .psect  $$afr1  rw,i,lcl,rel,con
        .psect  $$afr2  rw,i,lcl,rel,con
        .psect  $$buf0  rw,i,lcl,rel,con
        .psect  $$buf1  rw,i,lcl,rel,con
        .psect  $$buf2  rw,i,lcl,rel,con
        .psect  $$buf3  rw,i,lcl,rel,con
        .psect  $$buf4  rw,i,lcl,rel,con
        .psect  $$buf5  rw,i,lcl,rel,con

        .end
```

FCPMUBLD.CMD

```
;
;       f c p m u b l d . c m d
;
```

```
; build multi-user fllacp task (fcpmu) for rsx-11m
; this task image can then be munged with mubld to generate the images
; necessary for installation in an rsx-11m system
;       name - fcpmu
;       .odl - fcpmubld.odl
;       size - ??? kw
;       fcb list - 32. fcb's
;       separate directory buffer - 3 blocks
;
;
fcpmu/al/ac/mm/-fp/mu,fcpmu/-sp=
@fcpmubld.odl
;
; options input
;
task=fllacp
stack=32
units=1
uic=[1,1]
pri=149
par=fcppar:0:0
tskv=.sstvc:7
;
; the maximum size for this fcp is 12k. only the directory buffer, however,
; can lie in the last 4k (apr7).  the following buffers must be in the lower
; 8k (apr5 and apr6)
;
;
; allocate a separate buffer for index file bitmap
;
;       extsct=$$buf0:1006
;
; default is one buffer shared between bitmap and file headers (and directory
; blocks).  this action will speed up applications that do a lot of file
; creation and deletion.
;
;
;
; allocate a separate buffer for file headers
;
extsct=$$buf1:1006
;
; default from assembly is one buffer shared between fil headers
; and directory blocks (and index file bitmap).  this action will benefit any
; application that can afford the memory
;
;
;
; allocate private fcb pool space
;
; each fcb = 52 octal
;
;       extsct=$$afr1:xxxx
;
;       ,where xxxx = n * 52 (octal) and n = number of fcb's desired
;
```



```
extsct=$$afr1:2500
;
;
; *** note ***
; the preceding buffers and extsct's must lie in address space below apr7.
; do not extsct past 157777 octal!!
;
; the directory buffer below may lie, in whole or in part, within the
; address space of apr7. consequently, the maximum allowable for this extsct
; is <3.5k + n> or <17000 (octal) + n> where n represents the amount of space
; under apr7 that is still available after the preceding extsct's have been
; applied.
;
;
; allocate additional buffer space for directory blocks
;
;       extsct=$$buf3:xxxx
;
;       ,where xxxx represents 1000 for each 'additional' block desired.
;
; default from assembly is one directory block.  note that this
; action should be in addition to allocating a separate buffer for file
; headers.  allocating additional buffer space for directory blocks
; without a separate buffer for file headers does not accomplish very much.
;
extsct=$$buf3:2000
;
;
;
/
```

FCPMUBLD.ODL

```
;       f c p m u b l d . o d l
;
;       note:   this is not a real .odl file, but is referenced in the build
;               files in a line as @fcpmubld.odl
;
fcpmu
[1,24]fcp/lb:fllacp:fllcm:smcom:dispat:arwvb:overr:fcpgbl
[1,24]fcp/lb:fllbuf:dirbuf:init
[1,24]fcp/lb:alloc:blxio:cksum:clacc:cldac:cldir:crfid
[1,24]fcp/lb:darith:dlmrk:dracc:dreof:drex:drget:drini
[1,24]fcp/lb:drpac:drsef:drvlb:drwrt:dwpnd:fdrmv:fndnm
[1,24]fcp/lb:gtfid:gtmap:infcb:inwin:locat:mphdr:mpvbn
[1,24]fcp/lb:nxhdr:prock:rdhdr:rleas:rlfcb:rwvb:rwvbl:rwllb
[1,24]fcp/lb:scfac:scfcb:smalc:smnxb:smrvb:smscn:wacck:witrn
[1,24]fcp/lb:wrhdr:wtrnl:access:atctl:crfil:datim:deacc:entnm
[1,24]fcp/lb:excmp:excom:exten:ratcm:rdatt:rwatt:watcm:wratt
[1,24]fcp/lb:cldel:clatt:clcre:clext:clfcb:clnup:dlblk:dlfil
[1,24]fcp/lb:dlhdr:dralc:drcpy:drext:exthd:ixext:rmvnm:smdel
[1,24]fcp/lb:trunc:dmount
[1,1]exelib/lb,[1,54]rsx11m.stb/ss
/
```

A System for Technical Manuscript Preparation

Dan Dill
Chemistry Department
Boston University
Boston, Massachusetts 02215
617/353-4277

Traditional preparation of technical manuscripts is generally a slow, tedious, error-prone process. Secretaries are not uniformly familiar with special notations, and anyhow are hampered by reading handwritten text. Equations are difficult to type and once typed difficult to alter. Finally, because of the effort involved, one hesitates to make changes once a manuscript has been typed. These problems are largely eliminated by using a word-processing system with technical document capabilities. The facilities contained in the UNIX system (NROFF and TROFF) are best examples.

Unfortunately, most of the writers in the world do not also have a UNIX terminal or similar capability close at hand. However, the pieces (both software and hardware) now exist so that a modest and inexpensive but quite effective system for the preparation of technical manuscripts can be put together on PDP 11s. These pieces are the following:

- A VT52 or VT100 video terminal and a keypad editor such as K52 or KED. (Non-screen editors can be used, but are obviously much less friendly.)
- A NEC Spinwriter 5510/20 or 5515/25 with the Technical Math / Times Roman (TM/TR) thimble.
- The DECUS program RUNOFF, for document formatting.
- A RUNOFF preprocessor, such as the program TEXT described below, to facilitate use of the super/subscripting, alternate, and composite character set features of the Spinwriter.
- A macro preprocessor, such as MP, available as part of the DECUS C System, to allow local or system-wide definition of sequences of TEXT and RUNOFF commands as single commands, and to allow selective inclusion of text.
- A collection of command files to automate the processing through MP, then TEXT, and finally RUNOFF, to give final output on the Spinwriter.

The introduction of KED/K52 as part of RT-11 Version 4 and (optionally) for RSX-11M Version 3.2 make text entry so easy that RUNOFF becomes a viable alternative to a secretary. The introduction of the Spinwriter with the TM/TR thimble means that manuscripts of considerable technical complexity

can be produced in a single pass. The only missing piece is a program such as TEXT to extend the full capabilities of the Spinwriter to the RUNOFF user.

The Spinwriter print element can be positioned in 1/120-inch horizontal increments and 1/48-inch vertical increments, and so can construct precise composite symbols. The NEC TM/TR thimble has 127 (!) symbols, including many of the commonly used greek letters and special graphics. These capabilities of the Spinwriter are invoked by various, often lengthy, sequences of control characters. TEXT was created, as a preprocessor for RUNOFF, to spare the RUNOFF user the need to generate these sequences manually. At the same time, RUNOFF was modified to pass these non-printing command sequences raw to the output without affecting RUNOFF justification and fill computations. The resulting TEXT/RUNOFF combination provides a considerable (output-device-specific) extension to the RUNOFF command set.

1.0 What TEXT does

The TEXT commands are

```
↑ = superscript following character (or characters inside {···}),
| = subscript following character(s),
\ = print alternate/composite character(s).
```

The basic idea of TEXT is illustrated by the TEXT command string

```
↑1\p↑-|u.
```

This generates a string which, when processed by RUNOFF, will cause the Spinwriter to print

$$^{1}\pi^{-}_{u}.$$

There are bells and whistles, but this is the essential capability of TEXT.

2.0 Pulling out all of the stops

To illustrate TEXT/RUNOFF in action, there is on the next page a listing of a sample TEXT input file. (The example is taken from a manuscript by P. M. Dittman, Dan Dill, and J. L. Dehmer, entitled "Shape-resonance induced non-Franck-Condon effects in the valence-shell photoionization of O_2," that was prepared on the Spinwriter using TEXT/RUNOFF and submitted for publication to the Journal of Chemical Physics in March of 1981. The equations have been wrapped to successive lines to fit on the page. Tab characters are indicated by <TAB> and _ escapes TEXT commands. The resulting RUNOFF output on the Spinwriter is given on the following page.

Study of the *.TXT file and the resulting Spinwriter output illustrate many of the features of the TEXT/RUNOFF combination.

2.1 Example TEXT input file *.TXT. Output is RUNOFF input (*.RNO)

```
.nhd;.ps 58,255;.lm 5;.rm 80;.ts 14 77;.f;.j;.sp 1
Treatment of vibrational motion within the context of the adiabatic nuclei
approximation↑{\{24}} is accomplished as follows: In place of the fixed-R
dipole amplitudes given in Eq. (36) of Ref. 19,↑{\{25}} we define the
vibrational-state-dependent amplitudes
.tp 4;.rm 255;.nf;.nj;.b 2
<TAB>D↑{↑{(-)v|fv|i}}|{Lm|{\g}} = <v|f_|D↑{(-)}|{Lm|{\g}}_|v|i>
 = \IdR X|f(R) D↑{(-)}|{Lm|{\g}} X|i(R) .<TAB>(1)
.b 2;.f;.j;.rm 80
The vibrationally resolved integrated cross section is then computed
as↑{\{19}}
.tp 4;.rm 255;.nf;.nj;.b 2
<TAB>\s↑{↑{v|fv|i}} = (4\p↑{\2}/3)\ah\/\w  \S|{|{|{Lm|{\g}}}}
_|<v|f_|D↑{(-)}|{Lm|{\g}}_|v|i>_|↑{\2}  ,<TAB>(2)
.b 2;.f;.j;.rm 80
where h\/\w is the photon energy, and the vibrationally unresolved
integrated cross section is given by
.tp 4;.rm 255;.nf;.nj;.b 2
<TAB><\s> = \S|{|{|{v|f}}}\s↑{↑{v|fv|i}} = <v|i_|\s(R)_||v|i> ,<TAB>(3)
.b 2;.j;.f;.rm 80
where \s(R) is the fixed-R integrated cross section given by Eq. (41) of
Ref. 19.↑{\{25}} To obtain the vibrationally unresolved asymmetry
parameter, we begin with the general expression for the vibrationally
resolved differential cross section, which, in accordance with Yang's
theorem,↑{\{26}} is given by
.tp 4;.rm 255;.nf;.nj;.b 2
<TAB>d\s↑{↑{v|fv|i}}/d\W = (\s↑{↑{v|fv|i}}/4\p)
 [1 + \b↑{↑{v|fv|i}} P|{\2}(cos \T)] .<TAB>(4)
.b 2;.f;.j;.rm 80
Here, P|{\2}(cos \T) is the second Legendre polynomial, \T is the ejection
angle relative to the polarization direction of the light, and \b is
obtained, in analogy with Eq. (2), by substituting Eq. (1) for the fixed-R
dipole amplitudes in Eq. (40) of Ref. 19.↑{\{25}} Finally, using the
expression
.tp 4;.rm 255;.nf;.nj;.b 2
<TAB><d\s/d\W> = (<\s>/4\p) [1 + <\b> P|{\2}(cos \T)]
 = \S|{|{|{v|f}}}d\s↑{↑{v|fv|i}}/d\W<TAB>(5)
.b 2;.f;.j;.rm 80
for the vibrationally unresolved differential cross section, we can
express
the vibrationally unresolved asymmetry parameter as
.tp 4;.rm 255;.nf;.nj;.b 2
<TAB><\b> = \S|{|{|{v|f}}}\s↑{↑{v|fv|i}} \b↑{↑{v|fv|i}} / <\s>
 = <v|i_|\s(R)\b(R)_|v|i> / <v|i_|\s(R)_|v|i><TAB>(6)
.b 2;.f;.j;.rm 80
where \b(R) is the fixed-R asymmetry parameter obtained from Eq. (40) of
Ref. 19 using the fixed-R dipole amplitudes, Eq. (36) of Ref. 19.
```


2.2 Spinwriter RUNOFF output file from file *.TXT

Treatment of vibrational motion within the context of the adiabatic nuclei approximation[24] is accomplished as follows: In place of the fixed-R dipole amplitudes given in Eq. (36) of Ref. 19,[25] we define the vibrational-state-dependent amplitudes

$$D^{(-)}_{Lm_\gamma} {}^{v_f v_i} = \langle v_f | D^{(-)}_{Lm_\gamma} | v_i \rangle = \int dR\ X_f(R)\ D^{(-)}_{Lm_\gamma}\ X_i(R)\ . \qquad (1)$$

The vibrationally resolved integrated cross section is then computed as[19]

$$\sigma^{v_f v_i} = (4\pi^2/3)\alpha\hbar\omega \sum_{Lm_\gamma} |\langle v_f | D^{(-)}_{Lm_\gamma} | v_i \rangle|^2\ , \qquad (2)$$

where $\hbar\omega$ is the photon energy, and the vibrationally unresolved integrated cross section is given by

$$\langle\sigma\rangle = \sum_{v_f} \sigma^{v_f v_i} = \langle v_i | \sigma(R) | v_i \rangle\ , \qquad (3)$$

where $\sigma(R)$ is the fixed-R integrated cross section given by Eq. (41) of Ref. 19.[25] To obtain the vibrationally unresolved asymmetry parameter, we begin with the general expression for the vibrationally resolved differential cross section, which, in accordance with Yang's theorem,[26] is given by

$$d\sigma^{v_f v_i}/d\Omega = (\sigma^{v_f v_i}/4\pi)\ [1 + \beta^{v_f v_i}\ P_2(\cos\Theta)]\ . \qquad (4)$$

Here, $P_2(\cos\Theta)$ is the second Legendre polynomial, Θ is the ejection angle relative to the polarization direction of the light, and β is obtained, in analogy with Eq. (2), by substituting Eq. (1) for the fixed-R dipole amplitudes in Eq. (40) of Ref. 19.[25] Finally, using the expression

$$\langle d\sigma/d\Omega\rangle = (\langle\sigma\rangle/4\pi)\ [1 + \langle\beta\rangle\ P_2(\cos\Theta)] = \sum_{v_f} d\sigma^{v_f v_i}/d\Omega \qquad (5)$$

for the vibrationally unresolved differential cross section, we can express the vibrationally unresolved asymmetry parameter as

$$\langle\beta\rangle = \sum_{v_f} \sigma^{v_f v_i}\ \beta^{v_f v_i} / \langle\sigma\rangle = \langle v_i | \sigma(R)\beta(R) | v_i \rangle / \langle v_i | \sigma(R) | v_i \rangle \qquad (6)$$

where $\beta(R)$ is the fixed-R asymmetry parameter obtained from Eq. (40) of Ref. 19 using the fixed-R dipole amplitudes, Eq. (36) of Ref. 19.

3.0 Summary of symbols

The collection of all of the TEXT symbols and the currently defined composite characters available on the TM/TR font is given below. It is easy to add new composite symbols.

Greek characters

\a = α	\p = π		
\b = β	\r = ρ		
\D = Δ	\s = σ		
\d = δ	\t = τ		
\e = ε	\T = Θ		
\g = γ	\W = Ω		
\h = η	\w = ω		
\l = λ	\x = ξ		
\m = μ			

Special symbols

\f = ∫	\q = ∂		
\i = ∞	\u = ↑		
\j = ∝	\v = ↓		
A\:B = A·B			
a\.b = a·b			

Indexicals

\0 = ₀	\5 = ₅		
\1 = ₁	\6 = ₆		
\2 = ₂	\7 = ₇		
\3 = ₃	\8 = ₈		
\4 = ₄	\9 = ₉		

Overstruck (composite) symbols

e\' = é		A\% = Ā	
e\" = ë		O\/ = Ø	
e\! = e̊		A\A = Å	
e\~ = ẽ		E\U = Ë	
e\Y = e̲		e\V = ĕ	

Composite symbols

\+ = ±	\< = ≤	\= = ≃	\X = →
\- = ∓	\> = ≥	\Z = ≡	A|B = A‖B

Large composite symbols

\I = ∫	\[= [
\S = ∑	\] =]

4.0 OK. Where can I get one?

TEXT is written in MACRO 11, to run on PDP 11 computers under the operating systems RT-11 (Version 4) and RSX-11M (Version 3.2). There is a detailed manual describing the obvious and the less obvious features of TEXT, its operation, and examples of automated processing using the RT-11 command MUNG (and TECO) or using the RSX-11M indirect command processor task ...AT.. The package will be submitted to the DECUS PDP 11 library. For those in a hurry, it is available in the mean time on RX01/RX02 floppies from TEXT's author, me, (Al Franken ...) Dan Dill, Chemistry Department, Boston University, Boston, Massachusetts 02215, USA, telephone 617/353-4277. The modified RUNOFF will also be provided.

5.0 And so Virginia ...

We have been using this collection of tools for about seven months. Together they provide a critical mass that has markedly transformed into something almost pleasant the previously most unpleasant task of technical manuscript preparation. The cost above that of a typical RT-11 or RSX-11M installation is the price of a Spinwriter, about $2,500. In view of the return, this seems a small additional price to pay, indeed.

1980 - Spring - Chicago

[310,122] MTREK

This is a RSX-11M version of the Boeing real-time Star Trek game.

1980 - Spring - Chicago

[310,123] CHESS

This is a RSX chess playing program.

1980 - Spring - Chicago

[310,124] CHESS

This is a chess playing program written in "C".

1980 - Spring - Chicago

[310,125] ANOVAP
[310,125] BASS
[310,125] BG
[310,125] SHELL

This account has four entries of various types:

- o ANOVAP is a analysis of variance subroutine.
- o BASS is a program to design a bass speaker enclosure.
- o BG is a backgammon game.
- o SHELL is a shell sorting subroutine.

1980 - Spring - Chicago

[310,126] BIBLIOGRAHPY SYSTEM

This account has a bibliography system, written using TECO macros, to allow one to maintain and search a reasonable number of research papers.

1980 - Spring - Chicago

[310,130] DOD GRAPHICS

This account has a core graphics package written in "C" orginally for UNIX that needs some work for RSX environments.

1980 - Spring - Chicago

[310,131] BASIC GAMES

This account has many BASIC games written for the Micheal Reese BASIC program in [300,21].

1980 - Spring - Chicago

[310,132] ISC GRAPHICS

This account has some routines for the ISC color terminal. It includes a clock routine and an interactive drawing package.

1980 - Spring - Chicago

[310,133] ONTARIO HYDRO SUBROUTINES

This account has various subroutines from ONTARIO HYDRO. The routines include a database package, Fortran pointer routines (variable address handling), and other routines. Included are the following:

- o IGETAD stores the address of a variable.
- o INDGET gets the value stored at an address.
- o INDPUT stores a value at an address.
- o MOVDPI moves two consecutive words.
- o ICALL calls a subroutine using an address.
- o TINUM returns the terminal number and task name.
- o TIMOUT wait for 'n' seconds.
- o ERRMSG is a Fortran interface to MO....
- o BLOCK is a Fortran callable disk block read routine.

- o TRACE is an auto restart of task failure routine.

1980 - Spring - Chicago

[310,134] ESFLX
[310,134] DISPLY

This account has two programs. ESFLX is a program to transfer files between two systems using terminal lines. DISPLY is a program to convert various input formats to output formats, for example octal to real.

1980 - Spring - Chicago

[310,135] ALPHLIST
[310,135] HP PLOTTING PACKAGE

This account has two entries. ALPHLIST is a program to read, sort, and rewrite a directory. The HP PLOTTING PACKAGE is a graphics package for the HP 7221 plotter. It includes the following routines:

- o Initialization
  - * INITIA initializes the package
- o Page Definition
  - * OFFSET sets the orgin offset
  - * SIZE sets the page-axis size
  - * ORIGON sets the position of the plot origon.
  - * SCALE sets the size of the axis scales.
- o Cosmetics
  - * COLOUR sets the pen number.
  - * LINE selects the line number.
  - * SYMBOL selects the symbol number.
  - * SPEED sets the pen speed.
  - * TXTDEF sets the text size, slant.
  - * ANGLE sets the angle text is written at.
- o Background
  - * PLIFT lifts the pen from the paper.
  - * BORDER draws a border.

  - * AXES draws the X and Y axis.
  - * PABS moves the pen to a position.
  - * TEXT outputs the text line.
  - * TICKB outputs ticks around the border.
  - * TICKA outptus ticks on the axises.
  - * GRID draws a grid.
- o Foreground
  - * PLOT outputs a plot.
- o Plot Termination
  - * PAGE terminates a plot.

1980 - Spring - Chicago

[310,136] CONVERT
[310,136] ID
[310,136] MUSIC
[310,136] RDUNLMT

This account has four programs of various types:

- o CONVERT is a program to convert the record format of a file. For example, it can convert EBCDIC to ASCII.
- o ID is a MCR command to output the terminal and node number you are on.
- o MUSIC is a program to drive a 12-bit D/A and play music. The files for several Christmas songs are included.
- o RDUNLMT is a program to read unlabeled tapes and write the output to a file.

1980 - Spring - Chicago

[310,137] ACL

ACL is an accounting package for RSX-11M that collects various types of system usage information and writes this to accounting files. The account also includes programs to print reports from these files.

1980 - Spring - Chicago

[311,001] F4P
[311,001] KLAB
[311,001] MDIREC
[311,001] VPLOT7
[311,001] RSX11M
[311,001] VFLASH

This account has documentation prepared by the University of Texas about various parts of RSX-11M and their system. Included in the account are the following files.

- F4P is a summary of F4P (V2.5) and FOR language syntax.
- KLAB is a summary of K-series support routines.
- MDIREC is a summary of RSX-11M (V3.1) directives.
- VPLOT7 is a summary of the Versatec emulation routines.
- RSX11M is an introduction to RSX-11M.
- VFLASH is documentation on Vector General to Versatec hard copy.

1978 - Fall - San Francisco

[311,010] CALL
[311,010] PSHPOP

This account has two macros. CALL is a set of macros to generate Fortran compatible subroutine calls. PSHPOP is a set of macros to save and restore registers on the stack.

1978 - Spring - Chicago

[311,030] LOS ALAMOS SUBROUTINES

This account has a variety of subroutine modules. It includes some basic conversion routines, Tektronix 4010 graphics support, event control, and others. A partial list of the subroutines is as follows:

- ARGS will return the number of arguments in the Fortran calling list to the calling subroutine.
- ATTACH will attach or detach from a unit.
- BCD2B converts BCD to binary.

- BINBCD converts binary to BCD.
- BIN2D converts binary to decimal.
- BIN2O converts binary to octal.
- BLKFIL blank fills a string.
- B4010 initializes the Tektronix graphics package.
- CENTER centers a string of characters.
- CHOUT outputs a single character to the Tektronix 4010.
- CMDIN inputs a command line.
- CMDLK inputs a command line but does not reset it.
- CNVBCD converts BCD to binary.
- CSI is a Fortran callable interface to the CSI routines.
- CURSIS turns on the graphics cursor on a Tektronix 4010 and inputs the coordinates when a character is typed.
- DATESS outputs the current date and time to the 4010.
- DKREAD does logical reads or writes to the disk.
- ENDPLT terminates a graphics mode. It enters alphnumeric mode and centers the cursor.
- ERASE erases the storage scope.
- FETCH returns a character from a specified location in a string.
- FFMOVE moves characters.
- FFSCAN has free format parsing routines.
- GETCML is a Fortran callable interface to the Digital get-command-line routines.
- LABEL outputs axis labels to the 4010.
- LININ inputs a line of text from the terminal.
- LJUST left justifies a string of characters.
- MTAST allows a specified subroutine to be called after an interval of time.
- NCHAR will return the index of the last non-blank character in a string.

- o PUT will put a character into a string at the specified position.
- o REAST is a Fortran interface to the receive AST services.
- o RJUST right justifies a character string.
- o SCALE writes a numerical scale to the Tektronix 4010.
- o STROUT outputs a string of characters to the Tektronix 4010.
- o TPLOT outputs a vector line to the Tektronix 4010.
- o UICBIN converts ASCII strings to project and programmer numbers.

1978 - Spring - Chicago

[311,130] LOS ALAMOS SUBROUTINES

This account has command files to build the subroutine library in [311,30].

1978 - Spring - Chicago

[311,201] RSX-11D IND

This account has a version of the Indirect MCR command processor for RSX-11D.

1978 - Spring - Chicago

[311,205] SETTIM
[311,205] STATUS
[311,205] VOL

This account has three RSX-11D system programs.

- o SETTIM is program to read the time from a TCU-100 and set the system date and time.
- o STATUS is a program to display system status information on a Tektronix 4010 scope.
- o VOL is a program to output the mounted volumes to the user's terminal.

1978 - Spring - Chicago

[311,206] MTU

MTU is a magtape utility program designed to allow users to read, translate, list, and copy magtapes of various formats.

1978 - Spring - Chicago

[311,230] LOS ALAMOS SUBROUTINES

This account has command files to build the subroutine library in [311,30].

1978 - Spring - Chicago

[311,274] STARTREK

STARTREK is a version of the Star Trek game.

1978 - Spring - Chicago

[311,277] NOD
[311,277] PMP

This account has two RSX-11D status programs. NOD lists the usage of the system pool. PMP outputs a map of the partition usage.

1978 - Spring - Chicago

[312,001] RMSFTN
[312,001] SUPMAC

This account has two submissions. It has an enhanced version of SUPMAC. SUPMAC is a set of macros that provide structured programming constructs for Macro-11 code. RMSFTN is a set of Fortran callable subroutines to provide access to RMS ISAM files. Included are the following routines:

- o RMSOPE opens an ISAM file.
- o RMSCLO closes a file.
- o RMSCON connects a record access stream.

- o RMSDIS disconnects from a record access stream
- o RMSGET inputs a record.
- o RMSPUT outputs a record.
- o RMSFND finds a record.
- o RMSUPD updates a record.
- o RMSDEL deletes a record.
- o RMSRWD rewinds a file.
- o RMSRAC sets record access mode.
- o RMSKEY sets key information for record retrival.
- o RMSFRE frees a locked bucket.
- o RMSFLS flushes an I/O buffer.

1978 - Fall - San Francisco

[312,315] ASDEV
[312,315] BARON
[312,315] BD
[312,315] BIGTPC
[312,315] DDT
[312,315] DDT22
[312,315] DGT
[312,315] DOCTOR
[312,315] DSKFIX
[312,315] FOCAL
[312,315] FPEM
[312,315] GD
[312,315] GREP
[312,315] LISTRS
[312,315] MAP
[312,315] MSX
[312,315] RCONEW
[312,315] TPCDIR
[312,315] TPP
[312,315] FALL 1979 TAPE
[312,315] SPRING 1980 TAPE
[312,315] VDDRV
[312,315] XMITR

This account ,lds many different programs, primarily concerned with RSX-11D and IAS. Included in it are the following:

- o ASDEV is a RSX-11D device handler which allows a file to be associated with a terminal-like device. A command associates a file for either input or output. I/O which is then performed to the AS: device now goes to or from the file.
- o BARON is a game.
- o BD is a 16 channel MBD device handler.
- o BIGTPC is a modified TPC that can handle BRU-format tapes.
- o DDT is a symbolic debugger based on ODT. DDT22 is an enhanced version of DDT that can handle 22-bit addressing. Note, the newest version of DDT is virtual to the task and only requires a 200 word kernal and works for RSX-11M as well as IAS.
- o DGT is a program to read and write Data General RDOS tapes.
- o DOCTOR is a version of ELIZA written as a TECO macro.
- o DSKFIX is a disk block patching program.
- o FOCAL is a popular language.
- o FPEM is a RSX-11M floating point emulator.
- o GB is a RSX-11D null device that sinks all output and gives an EOF on all input.
- o GREP is a pattern search utility. It will search files for a specified pattern. This version is enhanced to include wild-card filenames.
- o LISTRS is a multi-column file lister program.
- o MAP is a RSX-11D utility to display a map of a partition.
- o MSX is a distributed operating system for coupled computers.
- o RCONEW is a utility to reset the EOF on locked files.
- o TPCDIR is a utility to list directories of TPC tapes. It can also extract single files from TPC container files.
- o TPP is a magtape utility that can position tapes, write EOF's and read and write card image files to and from the tape.
- o The TAPE DOCUMENTATION is annotated listings of the Fall 1979 (San Diego) and Spring 1980 (Chicago) tapes sets.
- o VDDRV is a virtual disk mechanism to provide encrypted disk structures.

- o XMITR is a network transfer program that uses the terminal driver. This version is conditionalized for IAS support.

1978 - Fall - San Francisco
1979 - Fall - San Diego
1980 - Spring - Chicago
1980 - Fall - San Diego
1981 - Spring - Miami

[313,001] BAK
[313,001] DSA SUBMISSIONS
[313,001] SEARCH

This account has three separate submissions.

- o BAK is an incremental backup utility and procedure.
- o DSA SUBMISSIONS is a description of the [313,*] submissions.
- o SEARCH is a TECO macro that performs pattern searchs on files.

1978 - Fall - San Francisco
1980 - Fall - San Diego
1981 - Spring - Miami

[313,010] TECO
[313,010] DSA TECO MACROS

TECO is a very versatile and powerful text editor. It is a character oriented editor and has many possible commands, including a full range of macro-type commands that allow TECO command files to be written and executed. This account has TECO V35 and various TECO macros:

- o ANAL analyzes teco macros to aid debugging conditional clauses.
- o COMPRES converts multiple spaces in a file to tabs.
- o EXPAND converts tabs in a file to the the equivalent spaces.
- o GETUIC stores the current UIC as a string '[ggg,ooo]' in q-register C.
- o REPALL search for a specified string in the files named and replaces with another string.
- o REVBLK updates module revsion histories. There are assoicated macros for fortran sources (FTNREV), macro sources (MACREV), and others.

- o SEARCH searches for a specified string in the files named.
- o SQU removes comments and other extraneous fields from Teco macros.
- o TYPE types files on the terminal.
- o VTEDIT implements a CRT screen editor using TECO.
- o VT52 implements a CRT screen editor for VT52 terminals.

1980 - Fall - San Diego
1981 - Spring - Miami

[313,020] TYPE
[313,020] AVD

TYPE performs the equivalent of a PIP TI:=filename but it is designed especially for terminal I/O. It will assume default file types and attempt several different ones, delete comments, and truncate trailing spaces. AVD is an enhanced version of the AVD in [346,100] that simplifies the command syntax and automatically assigns the first available virtual disk unit.

1980 - Fall - San Diego
1981 - Spring - Miami

[313,021] DVD

DVD is an enhanced version of the DVD in [346,100] that allows all disks assigned to a terminal to be deassigned. See also [313,2x] for other enhanced components of the virtual disk package.

1981 - Spring - Miami

[313,022] VDDRV

VDDRV is an enhanced version of the virtual disk driver in [346,100]. This version checks that the physical disk has not been removed. See also [313,2x] for other enhanced components of the virtual disk package.

1981 - Spring - Miami

[313,024] HELLO PATCHS
[313,024] BYE PATCHS