

**RSX**

**MULTI-TASKER**

September 1984 Issue

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	PDT
DECnet	Digital Logo	RSTS
DECsystem-10	EduSystem	RSX
DECSYSTEM-20	IAS	UNIBUS
DECUS	MASSBUS	VAX
DECwriter	PDP	VMS
		VT

UNIX is a trademark of Bell Laboratories.

Copyright © Digital Equipment Corporation 1984  
All Rights Reserved

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



FROM THE EDITORS

Allen A. Watson

1.0 ANAHEIM SYMPOSIUM DECEMBER 10-14

As I write this in August, I have just received the registration packet for the Fall, 1984 DECUS Symposium in Anaheim. You will probably be reading this in late September or early October, so the Symposium should be still two months in the future.

You therefore still have time to register, and this is a strong encouragement to do so. I particularly want to call your attention to the Pre-Symposium seminars that are sponsored by or of interest to the RSX SIG. There are several topics that should be really great to anybody truly interested in RSX.

The registration deadline for Pre-Symposium is October 30, so this is really the last time I can say anything in the Multi-Tasker to promote these seminars.

As mentioned in the last issue, the RSX SIG is sponsoring three seminars:

1. Industrial Automation: Tying Your Real-Time Device to RSX
2. Public Domain Software: An Indepth Review of the RSX SIG Tapes
3. RSX System Internals: A Developer's View

I think these three seminars are three of the best we have offered for a long time. The first topic, interfacing to unusual peripherals, is just made for Pre-Symposium. It's the kind of thing that would not work as a regular part of DEC Ed Services, but that is urgently needed by a small group of people who are faced with hooking up a special device. It won't be the world's most popular topic, but those who take it will probably feel it was the best investment in education they ever made.

The System Internals seminar is one I wish I could take even though I am no longer working with RSX on a regular basis. Listening to one of the RSX developers talk about how RSX works and why they did

it that way will, for anyone with an interest in operating systems, be just plain fun as well as being useful. This is a course for both beginning systems programmers and gurus alike. Brian McCarthy, the instructor, has been part of the RSX team for five years, and has worked with RSX for nine years. He is Principal Software Engineer with the RSX Development Group. You can't get a more expert person to teach a course like this. If you are interested, register at once; the seminar may be filled even as you read this.

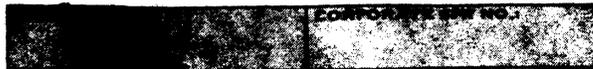
The seminar on the SIG tapes is the reason I can't be in Brian's seminar; I am one of the "instructors" in this seminar. This is going to be a dynamite seminar for any new or recent RSX user, or anybody who has never had a chance to spend time digging into the RSX SIG tapes. A group of seasoned RSX veterans who have spent hundreds or perhaps (between them) thousands of hours looking through and trying all of the free software that has been submitted over the past six years of SIG tapes are going to share, in depth, about what they have found and how to use it. And to make it a truly outstanding value, the attendees will each receive a tape containing all of the software to be discussed in the seminar -- a "Best of the RSX SIG Tapes" memorial album!

What will be given in this seminar, actually, is a series of mini-tutorials on Runoff, the LBL Software Tools, SRD, Portacalc, Kermit, CCL, and the DECUS compilers. Each tutorial is given by someone who has specialized, in one way or another, in the package he or she will speak about. Some speakers will be the actual authors of the software. You will never have a better chance to find out what the best stuff on the tapes is, and to have it all neatly packaged for you on a single tape.

The "best of" tape, I am certain, will be made available to all of you, whether you attend the seminar or not, through the DECUS library.



**SOFTWARE  
PERFORMANCE  
REPORT**



730841

✓ TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE \_\_\_\_\_ OF \_\_\_\_\_

OPERATING SYSTEM [ RSX 11M	VERSION V4-1	SYSTEM PROGRAM OR DOCUMENT TITLE DECnet 11M	VERSION OR DOCUMENT PART NO. V3-1B	DATE 25-MAY-84
NAME: J LEVINE FIRM: General Systems and Software.		DEC OFFICE AND CONTACT PERSON		DO YOU HAVE SOURCES? YES <input type="checkbox"/> NO <input checked="" type="checkbox"/>
ADDRESS:		REPORT TYPE/PRIORITY		
CUST. NO.:		<input checked="" type="checkbox"/> PROBLEM/ERROR <input type="checkbox"/> SUGGESTED ENHANCEMENT <input type="checkbox"/> OTHER		
SUBMITTED BY:		1. <input checked="" type="checkbox"/> HEAVY SYSTEM IMPACT 2. <input type="checkbox"/> MODERATE SYSTEM IMPACT 3. <input type="checkbox"/> MINOR SYSTEM IMPACT 4. <input type="checkbox"/> NO SIGNIFICANT IMPACT 5. <input type="checkbox"/> DOCUMENTATION/SUGGESTION		
PHONE:		CAN THE PROBLEM BE REPRODUCED AT WILL? YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>		
ATTACHMENTS MAG TAPE <input type="checkbox"/> FLOPPY DISKS <input type="checkbox"/> LISTING <input type="checkbox"/> DECTAPE <input type="checkbox"/>		COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? YES <input checked="" type="checkbox"/> NO <input type="checkbox"/>		
OTHER: PLEASE EXPLAIN IN PROVIDED SPACE BELOW.				
CPU TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE
				DO NOT PUBLISH <input type="checkbox"/>

Problem downline loading via DMP11AC/DMV11AC combination

A downline load of an 11/23 RSX11S system (V4-1) from an 11/44 host RSX11M V4 1 via a DMP/11 to DMV-11 link fails. The problem appears to be caused the routine FLOAT in the secondary boot returning a CSR of 160340 instead of the correct value of 160320. When FLAOT was patched to force 160320, boot proceeded correctly.

The documentation on DMP11, DMV11 and DECnet downline booting is very vague on the subject of CSR settings.

The "UNIT NUMBER FOR BOOTING" switch settings are not discussed at all. One can only guess at the function invoked.

*Submitted by Peter Bedwell  
DataLogic (Pty) Ltd.  
Blackheath, UK*

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.

SHORT NAME	MNT. CAT.	MNT. EXP.	REFER. EXP.
DATE RECEIVED IN AREA			

## Help Yourself

Raman Research Institute  
Bangalore, India

The Editor, Multi-Tasker  
C/O DECUS  
1 Iron Way  
Marlboro, Ma. 01752

Dear Sir,

I would like some help on the following matter

Our configuration - LSI 11/23 with the RSX-11M operating system. We have 2 RK05 disk drives. Each drive has one fixed and one removable disk, and each disk has two logical units, hence we have in all 8 logical units of 2.5 Mb capacity. The system image and the privileged tasks are on one logical and the utilities on another.

The problem which we encounter very often is that the MCR ACS command for allocating checkpoint space gives an error either (a) code -26 (no such file) or (b) code -36 (file id sequence no. check). When we examine the file header blocks using `DMP>TI:=[0,0]INDEXF.SYS/HF`, the 5th header block which is reserved for `CORIMG.SYS` i.e where checkpoint space is allocated has a different file ID (ie not (5,5)), a different file name and the number of retrieval pointers is also 0 in the case of error (a). Why does this occur and how do we avoid it? Because of this we have problems when tasks require checkpoint space. By the way, in our system option file we have not included checkpoint space allocation in the task image. Could it happen if the checkpoint file was in use and the system suddenly halted, either due to power failure or any other reason? We initialize a new disk and copy all the files into it and use it until the error occurs again. There is no other problems with our disks. All other files are intact.

Thank you,

Jayanthi Ramachandran

## MODEST PROPOSAL SESSION AT ANAHEIM

KMS Fusion  
3621 South State Road  
P.O. Box 1567  
Ann Arbor, Michigan  
48106-1567

August 3, 1984

To The Editor  
The Multi-Tasker

Dear Sir:

At the Fall DECUS meeting, we are again going to have a session called "Just a Modest RSX Proposal". This letter is a call for RSX users to participate in this session. The abstract is as follows:

Session Abstract

Let's face it, at one time or other each of us has thought (privately, to be sure!) the DEC did it all wrong, or really doesn't understand the needs of our particular interest group, or is missing the opportunity for a great product (hardware/software). Well, here is your chance to present to DEC and the RSX community Your Very Own Modest Proposal. Proposals should be limited to about 5 - 10 minutes in length. Viewgraphs, slides, movies or any other presentation aid are encouraged. The entries may be serious, or not so serious. A distinguished panel of DEC Developers, Product managers, and the attendees will judge the proposals in three categories; 1)"Noted" i.e., DEC likes it, 2)"UPG" i.e. DEC deserves it, 3)"Potential for Cult Following" i.e. the RSX community likes it.

The session is not designed to be a "gong" show, or a forum for beating up on DEC. It is, however, a chance for those of use who must exist in a Love/Hate relationship with RSX, to present those "If only..." ideas, or perhaps some thoughts that gently poke fun at DEC (remember UPG?). The success or failure of this session, is up to you. I have a few ideas (some serious, some funny) but certainly not enough to fill up a session.

Sincerely,

James Downward

Note From the Editor

I attended Jim Downward's Modest Proposal session in Cincinnati, and it was a really enjoyable time, with some serious things that came out of it. It could have been a little better, however, if more people had come prepared to present something, instead of improvising on the spot. So give some thought ahead of time, prepare a slide or two, and let us all know your favorite "neat idea" for RSX.

Working Group News

Jeff Hamilton  
Working Group Coordinator  
(214)457-4175

Date of this report: 10AUG84

The working group chairmen are as follows:

RSX-11M Unsupported Versions:

Bill Burton

Texas Research Institute

1300 Moursand

Houston, Texas 77030

System Performance and Accounting

Roy S. Maul

U. S. Air Force

HQ SAC / AD1AE

Offutt AFB, Ne 68113

DECUS Library

Bruce Zielinski

RCA

Marne Highway M/S 138-2

Moorestown, N. J. 08057

SIG Tape Collection

Glenn Everhart

RCA Government Systems Division

Route 38

Cherry Hill, New Jersey 08358

SRD

Bob Turkelson

NASA/Goddard Space Flight Center

Mail Code 962

Greenbelt, Maryland 20771

RSX Realtime

Ed Cetron

University of Utah  
Center for Biomedical Engineering  
3168 Merrill Engineering Building  
Salt Lake City, Utah 84112

Runoff

Chuck Spalding  
Adept Technology Inc.  
1202 Charleston Rd.  
Mountain View, Calif. 94043

Cheap Networks

Evan Kudlajev  
Philadelphia Electric Company  
P.O. Box 8699  
Phildelphia, Pa. 19101

The Unsupported Versions working group continues in its efforts to provide support for "old" versions of RSX. People presenting solutions to problems in previous releases of RSX will be submitting articles to Multitasker, covering changes to retro-fit Bonner lab Runoff, SRD, etc. on old versions of RSX. An article was sent to the Multi-tasker in regard to using BRU with MT: tape drives.

The System Performance and Accounting working group is continuing its work in preparing the index of the past RSXSIG tapes in regards to System performance and accounting. There is much interest in continuing the development of performance measuring tools. Further contact with DEC in regards the working groups goals is being pursued. An article is being prepared for the multitasker.

A restructuring of the tree for distribution of this tape-is being done. The people on the tree are being used in the evaluation of a sample 2400' 800 bpi tape that will be submitted to the DECUS library. The tape was sent to the volunteers the Friday before the Spring Symposium. Evaluation is being done on the tape at this time. Some responses have started to be returned. A letter has been sent to the Multi-tasker regarding the current work of the Library working group. Future submissions to the tape is being considered.

Glenn Everhart's report on activities of the SIG tape working group: "1. The SIG tape is close to distribution now. A preliminary version has been sent to the DECUS library and changes are now limited to updates and bugfixes with the Kermits. 2. The 2 cochairs of the PRO 350 subgroup have been sent the lists of interested people. They are:

Gene Hirschel  
9060 Palisade Ave. Suite 1011  
N. Bergen, NJ 07047  
201 662 9683 or 201 623 9825

and

Bob Uleski  
10 Truls End  
Fogelsville, Pa. 18051

215 264 2800 wk or 215 395 9018 hm

They are charged with organizing getting software off the tapes for PRO 350. The final tape will have several PRO 350 images in "hexified" form for easy Kermiting to the PRO. These include Kermit, TECO, copy, SRD, RNO, PortaCalc, Listrs, BASIC, Swedish PASCAL, MCR, a diskette mount/dismount, DTC, many of the tools from the DECUS C system, DECUS C kit (complete), graphics software, and more. If you have the mount utility online, you can mount PRO 350 diskettes foreign. A copy of the FLX.TSK image can be modified to handle RT format floppies then.

To do this I use DDT (also on the kit for PRO), built to allow debug of disk based images. I search for the string DX under a mask of 77577 (since the text is there with the high bit of the D turned on) and replace it with DZ, again with the high bit turned on. The text should appear in a table with lots of familiar device names. I also look a word later and change the 494. to an 800. for the size. Now I can use the bastardized FLX to read the foreign FLX disks. These changes convince FLX that the DZ device is a legal RT11 device (since DX is). I can't use that image on an RX01, but I don't expect I'll ever have an RX01 on a PRO. A similar trick works for RT format VD: disks.

I have the Midwest tree now and expect the others soon. The tape goes out as soon as they are ready. It will be available in the Library in BRU and VMS Backup formats, the latter for sites where the only tape drive lives on a VAX."

The SRD working group has started making changes to SRD for Micro/RSX and POS (basically for named directories). Hopefully they will have the changes made by the next symposium. A minor bug fix is being worked up for the Fall symposium tape. The SIG data base is being used in the evaluation of different SRD and SRD command file procedures.

The Real-time and Industrial Automation working group is working on the pre-symposia seminar that they are sponsoring. They are also working on a mailing about the role of the Real-time working group versus the LABS SIG. Other work is being done in the are of volunteers and objectives. Anyone with ideas, suggestions, or extra time and energy; please let Ed know.

The Runoff group has continued its effort to consolidate desirable features of several versions of Runoff into an "official" version. Chuck, due to work efforts, is finding the time to upkeep the working group more and more difficult. If there is anyone willing to help in this position, please get in touch with Chuck.

The Cheap Networks working group got off the ground with the volunteer effort of Evan Kudlajev who volunteered to be the chairman of this group. An evaluation of the current free software for networking applications is the group's first task. Further work in

the area of layering a mail application on this software was discussed. Evaluation of the current version of RSX KERMIT (2.17) is continuing. Distribution of the KERMIT kit is being done by KERMIT transfers and other virtual terminal transfer programs. An article is being prepared for the Multitasker.

The formation of the computer aided instruction working group still awaits the appointment of a working group chairperson.

If you are interested in providing information to a special working group concerning problems or ideas in that area, please get in touch with the working group chairman of that group.

### Crash Dump Analysis for Rookies

John McDonough

Dow Chemical  
633 Building  
Midland, MI 48640

Reported by John F. Reimer, Jr.

Subtitle:  
Crash Dump Analysis for Wookies

Do NOT be frightened ! Crash dump analysis is not as paralyzing as it seems. Granted, some crash dump analyses take weeks of intense scrutinizing to locate problem areas, but 85 to 95 percent of all crash dump analyses take only 5 minutes to solve. Given minimal background, it will please users to know that crash dump analysis is both self-sufficient and practical.

Before any analysis can begin, the actual crash dump must be down loaded to the scratch device. The key point here is to have a secure formatted device that can accomodate a dump of this type and size. Next, a "health" system is needed to do the analysis, so reboot the old one or find another one with the following 3 things. The three things needed for for a compatible system are 1. the same type of device the crash dump was taken on, 2. a version of the Crash Dump Analysis (CDA) program that matches the version of the operating system that ran on the crashed system, and 3. a file entitled RSX-11M.STB from a system with the same SYSGEN as the crashed system. (Note: BEFORE dumping with PDP systems, set the switch on the panel to LOCAL mode.)

After dumping the system, the Crash Dump Analysis Manual (CDA manual) and especially the PDP-11 processor handbook are needed. Save the executive listings and executive mappings for rare times when crash dump analysis is extremely complex. Print the analysis.

The first two places to check for indications to errors are the first and last pages of the printout. Start with the back page of the analysis, since the CDA can sometimes give you the solution itself. If not, follow the previously indicated manuals through the following steps:

- 1) Analyze the first page for obvious errors
- 2) Check the line "CPU ERROR = nnnnnn"
- 3) Check the lines of Pre-crash stating  
"PC = nnnnnn" (last Program Count)  
"PS = nnnnnn" (last Processor Status)

By working these particular statements through the reference manuals, you are on the way to getting the crash dump analyzed.

If the analysis is unsuccessful check the following points before proceeding; the System Commons, Pool areas and Pool linkage. Only after these steps should you sweat the monstrous task of seeking additional help or many nights of sleeplessness.

One final note regarding "HUNG" systems is to attempt to obtain more than one crash dump for a better chance of locating the obstacle or obstacles that crashed the system.

DECUS  
249 NORTHBORO ROAD  
BPO2  
MARLBORO, MA 01752

Attention: Multi-Tasker Edit

RE: Program submissions.

Dear Editors:

I make it a practice to read the RSX MULTITASKER on a regular basis. It is my major information source for happenings in the RSX world. In the last few publications I have found many grumbles over lack of material and concern over the future of RSX. This

communication represents my small contribution to help pump up the publication.

I looked over my 'toys' to find some that might be interesting to other users and decided to submit my version of the '...FRC' program and my '...WAK' program. I then discovered ...WAK wasn't ready for publication. This was mainly due to my use of some of our commercial fancy routines. I then had to re-write them to get a single source appropriate for publication. So much for lettering. Here are the product descriptions.

The ...WAK program functions as a alarm clock. It can be commanded to print out an announcement any time during the day. It's handy to tell you when your day is done, or to mark the time for an appointment. This task is currently running on RSX11M V4.0. The source is in MACRO 11. It uses the MUL and DIV instructions. ...WAK accepts a single command line containing text and the time to trigger. The command line is fairly free-form to allow you to place the time in an appropriate position. There are two acceptable time formats.

```
WAK :45 Sound the alarm 45 minutes from now.
WAK Sound the alarm when the time is 22:13
```

You can specify an absolute time delay, or a time of day. The task is limited to times after the current time, and before midnight. (So who works that late anyway?).

...WAK converts the time argument to the number of minutes remaining til alarm time. It informs the user how long that is and then delays. The alarm consists of repeated printouts of the command string with imbedded bells. The printouts contain a <CR> without a <LF>. This takes up a single line on your tube.

...WAK uses the RSX write break through (IO.WBT) I/O call to allow it to alarm on an attached terminal. This function requires a privileged task but doesn't require direct mapping to the exec. I believe the task is transportable and it's not necessary to re-build it on your system.

The ...FRC program allows one user to execute mcr commands on any terminal in the system. The idea for this program came from an early DECUS FRC task. This version differs in that it uses the modern RPOI executive call in place of the direct interference with the monitor used in the original. I don't have any original information to say who did the first version.

FRC accepts a single command line containing the terminal identification followed by an MCR formatted command to execute. The command format is:

```
FRC TTN:COMMAND
```

I usually use it to log off terminals for those persons who walk away with live tubes.

FRC TT12:BYE

FRC performs a limited amount of syntax checking to make sure the command has something resembling a terminal as the first portion of the command. It also contains a test to see if the terminal is our reserved target. This happens to be TT1:. If you want to change this feature the source can be easily changed. The use of the RPOI\$ function requires a privileged task but doesn't require direct mapping to the exec. FRC is currently running on RSX11M V4 with the RPIO\$ sysgen option. FRC is written in MACRO 11. I believe the task is transportable and it's not necessary to re-build it on your system.

The enclosed floppy contains the sources, task build files, and a copy of the task images.

Sincerely,

Frank Bosso

PRESCO  
Chief Engineer  
8 Lunar Drive  
Woodbridge, Connecticut 06525  
(203) 397-8722

#### Editor's Note

Source code for FRC and WAK follows:

```
.TITLE  FRC
.MCALL  QIO$C,QIOW$C,ASTX$$
.MCALL  WTLO$C,CLEF$C,SETF$C
.MCALL  RPOI$,DIR$,EXIT$$,ALUN$C
.MCALL  MRKT$C,GMCR$
.LIST   TTM
```

```
;+
; Task to simulate a mcr command on the indicated terminal
; Syntax is "FRC TTNN:COMMAND
;
; This source contains a stopper for TT1: as it is special
; on our system.
```

```

;
; *****
; * W A R N I N G *
; *****
; THIS IS A POWERFULL TASK.
; IMPROPER USEAGE CAN LEAD TO SYSTEM CRASHS
;
; By: Frank Bosso of Presco inc.
; (203) 397 8722
;-
FRC::      DIR$      #GMCR      ;FETCH A COMMAND LINE
MOV B      $DSW,R2      ;FETCH BYTE COUNT
BMI        FAIL        ;NO COMMAND GIVEN
MOV        #GMCR+G.MCRB,R1 ;POINT TO THE STRING
SPACE:     DEC        R2      ;COUNT A CHAR
BLE        FAIL        ;CANT FIND A SPACE
CMP B      (R1)+,#'      ;SCAN TO A SPACE
BNE        SPACE      ;NOT YET
CMP B      (R1)+,#"TT    ;VALIDATE INITIAL SYNTAX
BNE        FAIL        ;BAD START
CMP B      (R1)+,#"TT    ;VALIDATE INITIAL SYNTAX
BNE        FAIL        ;BAD START
SUB        #2,R2      ;ACCOUNT FOR "TT"
BMI        FAIL        ;COMMAND TOO SHORT
CLR        R5         ;INIT THE UNIT #
;
; Decode the response and send appropriate messages
; Via MCR... with the RPOI$ directive.
;
LOOP:      MOV B      (R1)+,R0      ;CALL A CHAR
DEC        R2         ;COUNT IT
BLE        FAIL        ;INPUT ERROR
BIC        #^C177,R0    ;STRIP PARITY
CMP B      R0,#':      ;TEST FOR EXECUTE
BEQ        DOIT
CMP B      #'0,R0      ;TEST FOR NUMERIC '0
BHI        FAIL        ;INPUT ERROR
CMP B      #'7,R0      ;TEST FOR NUMERIC '7
BLO        FAIL        ;INPUT ERROR
SUB        #'0,R0      ;MAKE BINARY
ASL        R5
ASL        R5
ASL        R5          ;MAKE ROOM
ADD        R0,R5      ;ADD IN NEW CHAR
BR         LOOP
FAIL:     QIOW$C      IO.WAL,5,1,,,,<ERRORM,ERRORS> ;SEND FAIL MSG
BR         BYBY
BADTER:   QIOW$C      IO.WAL,5,1,,,,<ERTERM,ERTERS> ;SEND FAIL MSG
BR         BYBY
;
DOIT:     CMP        R5,#1      ;CANT USE THIS TERMINAL
BEQ        BADTER
MOV B      R5,RPOI+R.POUN ;SET TERMINAL #

```

```

MOV      R2,RPOI+R.POBL  ;SET BYTE COUNT
MOV      R1,RPOI+R.POBF  ;SET BUFFER START
DIR$     #RPOI           ;SEND THE CMD
BYBY:    EXIT$$
;
RPOI:    RPOI$   MCR.....,5,1,,GMCR+G.MCRB,80.,200,TT,0
;
GMCR:    GMCR$                               ;The mcr command line getter
;
.NLIST BEX                               ;Dont wast paper
ERRORM:  .ASCII  /Can't decode the terminal syntax is "FRC TTNN:COMMAND"
        .BYTE   15,12
errors=-.errorM
ERTERM:  .ASCII  /Sorry, this terminal is reserved /
ERTERS=-.errorM
.even
.END     FRC

```

```

.TITLE  WAKE
.MCALL  QIOW$,QIOW$$,MRKT$,WTSE$$
.MCALL  EXIT$$,GMCR$,GTIM$$
.LIST   TTM
;+
; Task to set up a reminder for appointments.
; It accepts a time of day parameter and a string.
; Then it delays to that time and prints out the string.
;
; useage:
;
; WAKE 14:25 CALL GEORGE NOW
;      OR
; WAK :55 YOUR 55 MINUTE COFFEE BREAK IS UP
;
; By: Frank Bosso of Presco Inc. 21-Jul-84
; (203) 397-8722
;-
WAKE::   GTIM$$  #NOW      ;Fetch current time of day
DIR$     #INBUF   ;Fetch the line
MOV      #INBUF+2,R2    ;Fetch string start
MOVB    $DSW,R0       ;Fetch string length
BMI     ERROR        ;Must have a line
LOOP:    CMPB     (R2),#'0      ;Look for a time field
BLO     NOT         ; It's not a digit
CMPB    (R2),#':     ;absolute delay

```

```

    BEQ      AMIN          ;This looks like it
    CMPB    (R2),#'9
    BLOS    TIME          ; got a numeric
NOT:      INC      R2          ;Nxt char loc
    DEC     R0             ;Not this char
    BPL     LOOP          ;There are more chars
; Can't decode the argument, inform the user
ERROR:    JSR      R5,PRINT
    .WORD   MSGER          ;Cant decode
    JSR     R5,PRINT
    .WORD   INBUF+2        ; this command
    BR      BYBY          ;Get out
; A leading colon can mean absolute minutes
AMIN:     INC      R2          ; bump over the :
    CALL    ATOI          ;This must be a delta arg
    MOV     R0,MARK+M.KTMG
    BR      RDY
TIME:     CLR      MARK+M.KTMG ;Init to 0 min
    CALL    ATOI          ;Get hrs
    CMP     R0,#23.
    BHI     LOOP          ;Bah hour
    SUB     NOW+6,R0      ;Make delta hour
    BMI     ERROR        ;Wolt go past 24 hr
    MUL     #60.,R0      ;Make min
    MOV     R1,MARK+M.KTMG ;Plant it
; Hours converted, now look for minutes
    CMPB    (R2)+,#':      ;Min field
    BNE     RDY          ;No
    CALL    ATOI          ;Get min
    CMP     R0,#59.
    BHI     LOOP          ;Try further for a time
    SUB     NOW+8.,R0      ;Make delta min
    ADD     R0,MARK+M.KTMG ;Add to hours*60
RDY:     JSR      R5,PRINT
    .WORD   MSGDEL        ;Now say how long delaying
    MOV     MARK+M.KTMG,R0
    BMI     ERROR        ;No neg times
    MOV     #OUTBFR,R2    ; String buffer
    CALL    ITOA          ; Make total minutes
    JSR     R5,PRINT      ; Inform the user
    .WORD   OUTBFR
    JSR     R5,PRINT
    .WORD   MSGM
    MOVB   #15,INBUF+1    ;CR for overwrite
    DIR$   #MARK          ;Setup the delay
    WTSE$$ #1            ;Wait
; Time to wake up
RETRY:   DIR$   #ALARM          ; Try first print
    CMPB    $DSW,#IE.RSU&377
    BEQ     RETRY          ; Loop if the terminal is blocked
    MOV     #10,R5        ;Bug count
BUG:     DIR$   #ALARM        ; Now bug the user a while
    DEC     R5

```

```

        BPL      BUG
BYBY:   EXIT$$          ;Leave
;
; the timer packet
;
MARK:   MRKT$   1,1,3   ;Prototype is 1 min efn#1
;
; CURRENT TIME
;
NOW:    .BLKW   8.
;
INBUF:  GMCR$           ; The task command
        .BYTE   7,7     ;BELLS to wake you up
        .WORD   0,0     ;Print stopper
;
; A strong print statement
;
ALARM:  QIOW$   IO.WBT,5,1,,IOSB,,<INBUF+1,82.>
IOSB:   .BLKW   2
;
OUTBFR: .BLKW   40.     ;number holder
;
.NLIST BEX ;Don't wast paper
MSGER:  .BYTE   15,12   ;CRLF
        .ASCII  /Can't decode the time field /
        .ASCII  /use 'HH:MM' or ':MM' try again./
        .BYTE   15,12,0
;
MSGDEL: .BYTE   15,12
        .ASCIZ  /Delaying /
MSGM:   .ASCII  / m./<15><12><0>
.LIST BEX
;
.EVEN
.PAGE
        .SBTTL  ATOI SNCHK
;+
; Signed ascii to single precision signed integer.
; String pointer in r2, updated on return. Leading
; spaces and commas ignored. Conversion ends on first
; non-acceptable character. Result in r0. No overflow
; checks. All other regs preserved.
;-
ATOI:   MOV      R1,-(SP)
        MOV      R3,-(SP)
        CLR      -(SP)          ;Sign
        CLR      R0            ;Initial value
        CALL     SNCHK         ;Set sign
ATOI1:  MOV      (R2)+,R3      ;Get character
        CMPB     #' ',R3      ; space
        BEQ      ATOI1        ; skip
        CMPB     #',',R3      ; comma
        BEQ      ATOI1        ; skip

```

```

ATOI4:    SUB      #'0,R3          ;Extract number
          BMI     ATOI5          ;Negative if not numeric
          CMP     R3,#9.         ;Test upper range
          BHI     ATOI5          ; not numeric
          MUL     #10.,R0        ;Next digit
          ADD     R3,R1          ;Sum it in
          MOV     R1,R0          ;Back to dest
          BR      ATOI1
;
;end of input conversion
;
ATOI5:    DEC      R2              ;reclaim last char
          TST     (SP)+          ;Check sign
          BEQ     ATOI6          ; plus if equal
          NEG     R0              ;Make neg
ATOI6:    MOV      (SP)+,R3
          MOV     (SP)+,R1        ;Restore regs
          TST     R0
          RETURN                 ;Condition codes set.
;
;Routine to check sign for input number conversion
;
SNCHK:    CMPB     (R2),#'+'      ;Check plus
          BEQ     SNCHK1          ;Skip if so
          CMPB     (R2),#'-'      ;Minus
          BNE     SNCHKX          ;Exit if not
          MOV     #100000,2(SP)    ;Set sign on stack
SNCHK1:   INC      R2              ;Skip sign char
SNCHKX:   RETURN
.PAGE
          .SBTTL  ITOA    INTEGER TO ASCII
;+
; The 'itoa' routine converts a single precision
; integer in R0 to an ascii string in the output
; pointed to by R2.
;
; inputs:
;
; R0      single precision integer to convert
; R2      pointer to output buffer
;-
ITOA:    MOV      R2,-(SP)        ;Save orig output ptr
          TST     R0              ;Integer to convert
          BPL     ITOA.0          ;If plus, skip ahead
          NEG     R0              ;Make positive
          MOVB    #'-,(R2)+       ;Get sign char
;
; The conversion loop. We subtract a multiple of 10
; until the number becomes negative. Then we add the
; multiple of 10 back in once. We count the number
; of subtractions that produce a positive number
; to obtain the next digit to output.
;

```

```

ITOA.0:    MOV      #10000.,R4      ;Set largest multiple 10.
ITOA.1:    CLR      R1              ;Reset counter
ITOA.2:    INC      R1              ;Count subtract
          SUB      R4,R0            ;Sub multiple of 10.
          BGE      ITOA.2          ;Still positive?
          ADD      R4,R0            ;Fix number
          DEC      R1              ;Adjust digit
          BNE      ITOA.3          ;If not zero, then skip
          CMP      R2,(SP)         ;Any digits yet?
          BEQ      ITOA.4          ;If not, then skip ahead
ITOA.3:    ADD      #'0,R1         ;Make ascii digit
          MOVB     R1,(R2)+        ;Load digit
ITOA.4:    MOV      R4,R5          ;Copy mult. of 10
          CLR      R4              ;Clr hi order for divide
          DIV      #10.,R4        ;Get next lower ten
          TST      R4              ;Are we done?
          BNE      ITOA.1          ;If ne, then continue
          CMP      R2,(SP)+        ;Was number identically 0?
          BHI      ITA.5           ;If hi, then no
          MOVB     #'0,(R2)+      ;Insert "0" in buffer
ITA.5:    RETURN
.PAGE
          .SBTTL PRINT
;+
; Routine to print a ASCIIZ string
; on the default terminal
;
; Enter via
; JSR      R5,PRINT
; .WORD   STRING_ADR
; Return
; String output to terminal
; R0, R1 CLOBBERED
;-
PRINT:    MOV      (R5),R0         ; get string pointer
          CLR      R1              ; init char counter
PCNT:    INC      R1              ; count a char
          TSTB     (R0)+          ; till 'z'
          BNE      PCNT
          DEC      R1              ;Take off the excess
          MOV      (R5)+,R0       ; recall string pointer
; now print out the string lun 5, efn 1
          QIOW$$  #IO.WLB,#5,#1,,IOSB,,<R0,R1>
          RTS      R5              ;Back to caller
.END      WAKE

```

## RSX-11 and RSX-11M-PLUS Q &amp; A Session

Brian McCarthy, Gary Oden, Charlie Franks, Jason Gallant,  
Melanie Johnson, Joe Melvin, and Tony Lekas

Digital Equipment Corporation

Reported by John F. Reimer, Jr.

The question session pertained to many different aspects of RSX-11 systems. Subjects such as "POOL", "BRU", "EDT", and "EXECUTIVE CODE" were stressed by the general assembly and were, for the most part, answerable at the session by the panel of DEC experts. The session raised a many questions which will be included in the RSXSIG "Multi-Tasker". Within the "Multi-Tasker" will be the questions asked during the question and answer session, plus the solutions either given in response at the session or by the DEC expert panel after problem analysis.

To be inserted after Q's and A's...

To submit RSX-11M and RSX-11M-Plus problems to the DEC Software Maintenance group at Digital, please check to see that your question has not been answered at this session and published in the up-coming RSXSIG "Multi-Tasker", then submit a Software Performance Report (SPR) to DEC Software Maintenance.

Note that the response to the question and answer session was so over whelming that an additional session had to be added. Again, please refer to the RSXSIG "Multi-Tasker" for detailed questions and answers.

After Q and A write-up

Jim Webb Taulman Co. Atlanta, GA

Q: If and if so how is the COM-IOP-DZ device supported by TTDRV or 11M/11M+ ?

A: Not supported by 11M or 11/M+.

Jim Webb

Q: How do I generate timeout on a read that is less than one second (using a user written driver) ?

A: Must establish entry in clock queue.

Jim Webb

Q: How do you get Region starting addr and size of region in ILS ?

A: Get region parameters directive may return this information, otherwise get PCB address from ADB, then get reg start address from PCB. ( all via privileged task )

George White McDonnell Douglas

Q: We have had some problems with the task builder running on a PDP-11/70 RSX-11M-Plus 2.1 (STK v 4.10 and v 4.12) One symptom is that large structural analysis tasks no longer build because the slow task builder runs out of resources. The STK dies with the message "NO VIRTUAL MEMORY STORAGE AVAILABLE." The second symptom is that a task built with the v4.12 STK did not run. It appears that the autoloader vectors were not set correctly for the overlay ( the task built and ran correctly with STK v4.02). The third symptom is with an overlaid task compiled by FORTRAN-77. The task built but did not run. However, when compiled with FORTRAN-4 Plus, the task built and ran.

A: Please document the problem through the usual channels (send in SPR)

Larry Solomon Kulicige & Soffa Ind. Horsham, PA

Q: After doing a SYSGEN with a PDP-11/70 processor with (3) RM03's UDA-50 (With RA80 + RA81), RX02, TEL6, 64 lines on DZ's which adds a TU80 and 16 more lines, then SAV hangs. Before the SAV, the soft boot worked and the system could "CON ONLINE ALL". What went wrong ?

A: Unknown. Send in MAGtape of unbooted software and corrupted (Half saved) system.

John Smedinghoff RCA Consumer Elec. Div. Indianapolis, IN 46206

Q: (RSX-11M+ v2.1) I have an 11/44 system with 1920K words of memory. Why doesn't my baseline system boot ? Shouldn't any bug be well documented and the documentation distributed with each RSX-11M+ kit?

A: No Answer.

John Smedinghoff

Q: (RSX-11M+ v2.1) If I have an indirect command file consisting of the following 2 lines,

.DISABLE DISPLAY, QUIET

;This line should PRINT then why doesn't the second line print ?

A: Has been fixed in update.

UnKnown

Q: Running v4.1 of RSX-11M on an 11/23, trying to boot a 22 bit system from an 18 bit one, we get hung at the very end of the system. The last command, ">G" is processed, but there is no response to commands after that. If we boot for 18 bits, then say "RUN \$SHUTUP", we get the message "SHUTUP's TI: is a PSUEDO DEVICE." Are these related ? How can we solve them both ?

A1: Sounds like one that occurred in configuration of exactly 128K- but we thought it was fixed in version C.

A2: This does not seem related, but not sure - we don't have an answer.

UnKnown

Q: Is it possible to make CO:A file rather than a terminal device ?

A: Yes - use SET/COLOG

Darwin Geiselbrecht Tennessee Gas Pipeline

Q: While running M+ v2.1 and update B, we are getting crashes with an odd address trap. CDA always shows PIP as the active user task. Is this a common problem that was fixed by update C ?

A: Problem has not been encountered. SPR.

UnKnown

Q: For the 3 "Flavors" of RSX, what is the smallest distribution media, plus SYSGEN media ?

A: 11S RL01

11M RL01

11M+ RK07 TO GEN

RL02 WITH PREGEN

Jim Neeland

Q: KERMIT-11 using RMS 2.0 for file operations, has a HELP file in LB:[1,2]. Works fine until installed non-checkpointable, then RMS returns FILE-NOT-FOUND. Install with FULL INCR does not fix problem.

A: Check GSA routine - see what it is using for determining available space. Wrong if using .LIMIT.

UnKnown

Q: How can you get free disk space in a form that can be teted in an indirect command file. PIP will not write it to a file.

A: Use VFY

Rolf Kamp JEOL 235 Birchwood AV. Cranford, NJ

Q: On an RSX-11M system, is there any method in which you can tell who has a device mounted ?

A: DEV/DR: will say what is mounted but not who (unless allocated).

Rick Sharpe Toledo Edison

Q: Is ther an easy way to print trailer pages with the QMG.

A: No easy way at this time.

UnKnown

Q: (No QIO timeout, PDP-11/34 RSX 11M v3.2) Device does not detach on graceful exit or abortion. Are these problems in updates to v3.2 or do I need v4.0 or v4.1 ?

A: We have not seen the problem.

UnKnown

Q: How do I flush type ahead buffer in QIO flush ?

A: Set characteristics control of terminal on flush.  
(Audience: control might do it...)

Keith Murphy W.R. Grace 100 N.Main Building Memphis, TN 38103

Q: After changing from v4.0 RSX-11M to v4.1 we encountered a problem during the "POOL EXHAUSTED" condition - The system HALTS ! We found that the lower part of memory from addresses 0 to 60(octal) had been zeroed. We tried running without PMT, and our boot block on the system disk got wiped out the next time POOL ran out.

A: Submit an SPR.

Donald Criley Cheveron Research CO.

Q: Why do the Purclue ISA subroutines come in the form of 4 concatenated object files some of which contain modules with the same identical name and ident yet derived from different source code?

A: No answer, will look into documentation.

Michael Schilling Cheveron Research CO.

Q: In M+ v2.1, all terminal driver output is buffered, large output buffers can exhaust POOL. Why is this allowed, and why is output to DH11's ever buffered ?

A: Ouptut to DH11's must be buffered to do tab expansions.

Yural Brash Brash concepts 152-18 Union Tpke Hillcrest, NY 11367

Q: What is the status of the following documents: RSX-11M/M+/U/S and Logic manual (internals) ?

A: NOTED !

Paul Simons Convex Convex Southington, CT

Q: What do the deletion of dynamic regions and the I/O count (T.IOC) of a task have in common ?

A: The dynamic region may contain an I/O buffer.

Paul Simons

Q: What advantages do ALSCT and RLSCT provide over managing the memory yourself ?

A: Don't worry about them. Use virtual arrays. They were provided to impliment semi-virtual arrays in earlier versions of FORTRAN.

Paul Simons

Q: Is there any way to recover memory that is not used in the I/O page? (If we only have three devices, why tie up 4098 words?)

A: No. Different processors use I/O page in different manner.

Jasper Davis Teradyne, Boston, MA

Q: mou dd:/our on M/M+ what does /our do besides subvert label chak and user already mounted check?

A: No, does not do anything but skip check on volume label.

Ted Niemann SCM Corp Cleveland, OH

Q: F77 program (5.0) running under M+ 2.1 update B. When system is busy, the Fortran program can cause the system to hang, i.e. RMD stops updating screen and then terminates.

A: Hardware problem will PIRQ register. Use ERRSRT to report floating over and underflow.

Kurt Hillig Univ. of Michigan

Q: I'm trying to write to a graphics terminal. If I use a Fortran write statement it works bu is slow. If I call WTQIO with the "WRITE-PASS-ALL" code, then when the terminal sends an XOFF to the PDP, after about 5 seconds I get "LOW POOL" message. Why and how do i fix it?

A: Don't use "WRITE-PASS-ALL" Try an ordinary write-logical-block. IO.WPA turns off the terminal synchronization. If Fortran write works then "WRITE-PASS-ALL" is not needed.

Alder Schmitd (?)

Q: Micro PDP-11/Micro RSX seems like it was designed for user added devices i.e. loadable user written driver, connect to interrupts. This is not good when used in an industrial control enviroment because of possible enexpected motion on industrial machine. Comment, please.

A: RSX Architecture book denotes used and therefore checked portions of I/O page.

UnKnown

Q: Bye aborts SYSLOGOUT.CMD when system is loaded. WHY?

A: Rebuild Bye with longer timeout.

UnKnown

Q: 11/23 traps to ILIO (<= ???) during selftest continues ok, will not auto configure. Y?

A: Bus Refluctiors? / Parity memory.

Robert Grimm The Milton CO.

Q: (RSX 11M+ v2.1 PDP 11/24) Is there a task or amethod of analyzing the contents of POOL ?

A: Check the Fall or Spring 1983 SIG tapes for O.P.A. online POOL analyzer.

Eric F.G. Wilson Dofasco Hamilton, Ontario

Q: (RSX-11M+ v2.0B on a slaved terminal) A Task attaches the terminal, has aread pending to that terminal:

CON OFFLINE THAT TERMINAL

CON ONLINE THAT TERMINAL if you now hit a key on that terminal the system crashes.

A: SPR.

E. Wilson

Q: Is their any SIG utilities that will online compress a disk.

A: No answer. (Audience: unless files were restricted to be contiguous with no extends by system.

Wilson

Q: RSX-11M+ v2.0B; TT-DRV - set terminal remote -when you lift the lid or get paper out of LA120 is now set no formfeed.

A: Look into TTMOD - some conditionally assembly code can be enabled not sure when or what it is.

Wilson

Q: CBD REGION/TASKS will only show active tasks attached to a region it will not show non-active installed tasks that are RESCOMED the region.

A: It is on the RSX Wishlist...future stuff

Ed Cetron Cntr for Biomedical Design; 3168 MEB, Univ. of Utah; Saltlake City, Utah 84112

Q: Why oes SAVE only let you SAVE a system image only once. Further SAVES require a new ".SYS" file.

A: Fixed in update C.

Jasper Davis Teradyne Boston, MA

Q: Mount DYN: who does not have its device cable attached, causes an executive Crash. In Las Vegas this was supposed to have been fixed by autopatch B for M+ v2.1 and M v4.1, however if configure is run "ALL" then configure attaches the DYN: and seems to severt the fix included in autopatch B. Is this fixed now?

A: Not fixed until Autopatch C or D, not sure which.

#### CONVERTING RSX-11M FORTRAN PROGRAMS TO VMS

Submitted by Ed Cetron  
Author: George Metos  
ECN Corp  
1104 East Ashton Ave.  
Salt Lake City, Utah 84106

The following is a list of the problems and solutions encountered in one site's conversion of Fortran programs written on a PDP-11/34 to a VMS environment (there would be absolutely no difference whether the target computer were a 750, 780, or 782, etc.). RSX-11M V3.2 and VMS V3.0 were used.

The typical software at this site was business oriented, with emphasis on accounting, data entry, and reporting. There was very little "non-standard" coding, i.e. code that would require programmers to be proficient in areas other than that of applications programming. Several functions applicable to one PDP-11 are not even available on another PDP-11, but on this site, it was not of concern. Note that some of these items apply only to Fortran, while others might apply also to PL/1, C, Basic, and other languages.

1. Logicals, Accounts : RSX typically refers to its hardware directly, such as DRA0:, LPA0:, etc. This is acceptable to VMS, but much flexibility can be gained by assigning a logical (such as SYSSYSDEVICE: or DISK\$PROGRAMS) to the particular device. Entire accounts can then be moved around with only the need to reassign the logical name. This is also helpful when moving programs from multiple RSX disks to one larger VMS disk. For example, DRA0:, DRA1:, DMA0:, etc. can all be assigned as DISK\$PROGRAMS now if desired.

RSX accounts were of the [44,100] UIC variety, VMS can also handle this with one minor change...they must be changed to the 'name' variety, in this case [044100]. VMS allows the use of [SUSAN] style of account names, and can also equate the two accounts [044100] and [SUSAN] to be the same account (just the file directory will be duplicated).

2. Files : VMS offers a much richer variety of options available to open and close files. Some of the older RSX programs used 3 separate statements to open files. VMS utilizes one open statement that allows files to be accessed much more elegantly. The old statements will still be compatible with a few minor modifications. Record lengths, for example, are still expressed in integers, but VMS works with 32-bit integers as opposed to 16, so accordingly, record lengths need to be halved in the open statement. RSX had an unfortunate habit of leaving files in an indeterminate state if crashes occurred, or if users tried to access the same file. VMS handles this much better, but only when its own open statement is used, therefore you have much to gain by changing old open statements to the new type. The VMS close statement has even more power. Files can be deleted upon close, queued to a printer, ect., giving command files, then the terminal unit number (usually 5 or 6 under RSX), must all be unit 6...this is because SYS\$INPUT and SYS\$OUTPUT are both assigned to the terminal from the DCL level, but suddenly change when a program is run from other levels (such as from command files).
3. Variable types, sizes : Use FORTRAN/NOI4 to compile your old programs if it is important for integers to be 16 bit, otherwise all integers default to 32 bit. Comparison of byte variables that worked in RSX may not work in VMS unless they are declared to be LOGICAL\*1. For example I=IAND(Byte,2) will give VMS some trouble, whereas RSX will handle it just fine. RSX also had a somewhat different algorithm to evaluate equations. For instance to evaluate REAL=I1+I2, RSX would have no trouble if the total were over 32767, but VMS would generate an integer overflow!, so some manipulation may be necessary. VMS also has less mercy for problems associated with array out of bounds, and when converting multitudes of programs, one is sure to find errors that RSX ignored, that VMS would not.

4. Miscellaneous statements : Common statements may have to be repositioned in VMS because the two compilers (RSX and VMS) assign variable types using "slightly" different algorithms. Continuation characters may have to change, because VMS may have trouble with special characters like '\*' used in such applications, it likes numerals. Some of the error numbers will change slightly, this is only a problem if action is taken on the value of the error, the manuals will give you an idea on the differences.

In general, VMS will handle just about anything at all from Fortran. For example ISTATUS=LIB\$SPAWN ('RUN PROGRAM2') is acceptable in VMS. RSX could handle some spawning, but VMS cleans up the process extensively and allows virtually anything that can be done from DCL level to be done in any high level language.

Anytime someone attempts this type of conversion, they will undoubtedly encounter many of these items, and also generate a few of their own. Once converted to VMS however, there is usually no desire to return. VMS is a huge improvement.

# Software Catalog Under RSX-11M

14 South 1 PPG Place  
30 April 1984

Dear Multi-Tasker Editor:

Maintaining a software catalogue under RSX11M can be a frustrating experience:

- o Have you ever waded through large files or directories containing scores of small files, trying to determine what code belongs together, so you can cannibalize logic -- or debug it?
- o Have you ever tried to port a system to another site, only to discover that the uics that you used to develop your code were already occupied?
- o Have you ever longed for a directory structure that tells you something about your software's hierarchy?

I want to share some ideas on managing software development which grew out of a three-year effort to organize and catalogue a large number of interrelated systems under RSX11M. I've developed three techniques for coping with the three problems enumerated above.

## I. Some Ideas on Packaging Programs under RSX11M

I try to package my programs according to the following guidelines which, according to proponents of structured design, cut system development time and enhance code maintainability and portability:

- 1.) partition programs into small, well-defined modules (a page or two, maximum, with one subroutine per module).
- 2.) make each module a separate file
- 3.) segregate related collections of modules
- 4.) limit the number of items in any collection to 7 +/- 2 maximum, a magic number which limits quick human comprehension

I believe that most RSX11M programmers develop software according to similar guidelines, but compromise system portability and maintainability by packaging software in either of two ways:

- a.) "le grand file": dump code into big files.
- b.) "le grand directory": dump files into mammoth directories.

The "big file" approach tends to violate guidelines 2, 3, and 4. The "big directory" approach tends to violate guidelines 3 and 4. My approach has been to stop regarding directories as sacred cows, i.e., to use directories to help partition my code:

- o Modules are grouped into families. I group modules together which manipulate a well-defined item; e.g., one family may consist of gets/puts to a given header record.
- o I limit families to 7 +/- 2 modules, maximum. If more modules start to creep into a family, I rethink the classification.
- o Each family is assigned a separate directory.
- o Each program main module is assigned its own directory.
- o A collection of families which manipulates a distinct collection of items is treated as a system, and assigned to its own group. My basic guideline is, should the collection have its own manual?

One reason, perhaps, why programmers package modules into large files is that increasing the number of program fragments for a given program increases the difficulty of maintaining that program's command files. I minimize command file maintenance by referencing routines on a per family rather than on an individual basis:

- o I assume one object module library, FILES.OLB, for each family of routines.
- o Since each module family occupies its own uic, the command file which creates FILES.OLB only needs to know the target uic and the source file type (e.g., "\*.MAC"; see UICLIB.CMD, enclosed with this article).
- o Similarly, all program-building command files reference the FILES.OLB libraries by uic rather than specific modules by name.

I only need to edit my program-building command files when I create a new task, or when I need to add a module family to or delete a module family from my taskbuild specification.

Note: The problem with using archivers to reduce directory clutter under RSX11M is that the only DEC-supported archiver, LBR, does not allow "\*.\*" extraction from text libraries (.ULB files).

## II. Logical Uics

I prefer that my code be as free from uic dependencies as possible, to make it easier for me to shuffle code between uics as needed, e.g.,

- o to resolve uic conflicts while copying software between volumes
- o to reflect changes in program hierarchy (see III., below)

To minimize drudge work associated with uic changes, I have devised a method for implementing "logical" uic names for a given RSX11M volume. I use the term "logical" it parallels DEC's distinction between "physical" and "logical" devices. The method involves six major steps, as follows:

- 1.) Assign a unique two-character name to each active uic group on the volume. For example, "PT" may designate a group dedicated to paper-tape-transfer software.
- 2.) Assign a unique six-character name to each member uic within each active group, as follows:

- 1st character: "\$" (...AT. convention for global variables)
- 2nd, 3rd char: the group designator
- 4th, 5th char: a two-character member designator, unique within the current group
- 6th character: "U" (my own convention: denotes uic name)

These names, which I refer to as logical uic names, have been defined so that they double as indirect command file processor global variable names. For the sake of the example, "\$PTESU" might define the member uic in the PT group which contains "emit final program status" routines.

- 3.) Create a master command file under [1,1] which -
  - o accepts one parameter: a two-letter group designator
  - o initializes all logical uic names for the specified group, setting each designator to a string defining its proper uic

For example, if I pass this command file "PT" as a parameter, the symbol "\$PTESU" should be defined on return as "[100,20]", if [100,20] is really where the paper-tape-transfer-emit-status family of routines is currently sitting.

I like to implement this command file as a two-level command file:

- a. The master command file knows all the legal two-character designators for its volume, along with each designator's group code. In the above example, the master command file would be responsible for knowing that the PT software currently resides under group 100.
- b. Each active group has its own command file, invoked invoked with one parameter: physical group number. These group-by-group command files are then responsible for completing the logical-to-physical uic connection, using statements like:

```
.SETS GRP P1
. . . . .

.; Error message subroutines
    .SETS $PTEMU "[GRP',70]"

.; Emit status routines
    .SETS $PTESU "[GRP',100]"

. . . . .
```

Note that these command files' names must be derivable from the group name: e.g., "[1,1]PT.CMD" would do for our current example. In practice, the names should really be a function of both group and volume name to promote transportability across volumes.

- 5.) Indirect command files, by convention, may only reference logical uic names. Command files establish correspondences between logical and physical uic names at execution time by invoking the master uic lookup command file under [1,1].

As an example, suppose that a program-building command file needed to reference the paper-tape-transfer-emit-status subroutine family. If the master uic-defining command file for the current volume were named "VOLUICS", the program-building command file first executes

```
.; Establish uic names for the paper tape group
    .IFNDF $PTESU @[1,1]VOLUICS PT
```

and then references '\$PTESU' as needed.

- 6.) Programs which reference uics are coded using "hooks" which permit indirect command files to establish uic names at program creation time (using GBLPATs, or EDT initializer files, or whatever).

Using the above scheme, the only file on a given volume that's bound tightly to a specific uic is the master lookup table under [1,1]. If I need to move directory within a group, I adjust the corresponding group lookup table (e.g., [1,1]PT.CMD). If I need to move an entire group, I adjust the master lookup table (e.g., [1,1]VOLUICS.CMD) and use my GRPCOP command file to do a rename across groups (see GRPCOP.CMD at the end of this article; PIP does not support uic-by-uic transfers across groups).

### III. Hierarchical Directory Structure

The major payoff that comes from implementing logical uics is the freedom to use directory numbers to model program hierarchy with minimal hassle due to directory reshuffling. For example,

- o Within a group, assign modules lower in the hierarchy to lower-numbered members; i.e., routines in [100,4] may invoke routines in [100,2] but not in [100,6].
- o Within a volume, assign systems lower in the hierarchy to lower-numbered groups; i.e., routines in [100,\*] may invoke routines in [77,\*] but not in [101,\*].

### IV. Afterword

Some people with whom I have spoken question whether catalogues developed using these techniques are too complex or too cumbersome to maintain. My experience with this system has been very good. Currently, I'm managing twenty-one interrelated systems on a single RL02. Implementing the system was straightforward; others who cannot type as quickly or who are not as proficient with indirect command files may have more problems. Managing the volume has been much easier since the system has been in place. Finally, I do find that code in the catalogue has been easier to maintain and reuse.

I would be very interested in comments from other members of the RSX community.

Respectfully,

A handwritten signature in cursive script, reading "Phillip Pfeffer".

Phillip Pfeffer

systems analyst  
PPG Industries  
Glass Engineering Group

```
.ENABLE GLOBAL
.ENABLE SUBSTITUTION
```

```
.;+++++
.;
.; UICLIB.COMD - Create object module library from uic source files
.;
.; Create an object module library from .MAC files under one Fox text
.; transfer group uic. Invoke this command file as follows:
.;
.; @UICLIB (+[sw]) (uicspc)
.;
.; where:
.; parentheses indicate an optional argument
.; [sw] is any combination of the following -
.; "L" - causes assemblies, compiles generate listings.
.; (default: no listings)
.; uicspc - uic for which to generate library
.; (default: current uic)
.;
.; FX: assumed for sources, listings, objects and object module library
.; "FILES.OLB;1" assumed for object module library name
.;
.;-----
.; *****
.; * Preliminary processing - determine command file parameters *
.; *****
.;
.; *** ** Decode caller switches *** **
.;
.; .SETF LSTS
.; .SETS SRCU P1 ! assume no switches at first
.; .TEST P1 "+"
.; .IF <STRLEN> <> 1 .GOTO SWDONE
.; .SETS SRCU P2 ! switches present
.; .TEST P1 "L"
.; .IF <STRLEN> <> 0 .SETT LSTS
.;SWDONE:
.;
.; .TEST SRCU ! check if uic was specified
.; .IF <STRLEN> = 0 .SETS SRCU <UIC> ! no - default
.;
.; *** ** Define working device *** **
.;
.; .SETS DEV "FX:" ! working source device
.;
.; *** ** set up file synonyms *** **
.;
.; .SETS MACLST DEV+SRCU+"MACLST.TMP;1" ! PIP (temp) list file
.; .SETS MACTMP DEV+SRCU+"MAC.TMP;1" ! macro (temp) cmd file
```

```

.SETS  OLBFI  DEV+SRCU+"FILES.OLB;1"  ! output object library

.SETS  LBRTMP DEV+SRCU+"LBR.TMP;1"    ! lbr (temp) cmd file
.SETS  CLNTMP DEV+SRCU+"PIP.TMP;1"    ! cleanup (temp) cmd file

.;  ***  ***  Init assembler command switches  ***  ***

      .SETS LSW  "/NL:BEX/-SP"
      .SETS OSW  "/EN:LC"

.;  *****
.;  *  Create command files from uic source files  *
.;  *****

.;  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
.;  *  Init object module library create command file  *
.;  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *

      .OPEN  #0  'LBRTMP'
      .IF <FILERR> <> 1 .GOTO FE0000          ! set up file create
      .DATA  #0  'OLBFI'/CR:30:64:64
      .IF <FILERR> <> 1 .GOTO FE0000

.;  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
.;  *  Init object cleanup create command file  *
.;  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *

      .OPEN  #1  'CLNTMP'
      .IF <FILERR> <> 1 .GOTO FE0010

.;  =====
.;  !  Process .MAC files under UIC  !
.;  =====

.;  ===  ===  Init - create list of .MAC files under UIC  ===  ===

      PIP 'MACLST'='DEV''SRCU'* .MAC/BR

.;  ===  ===  Init - open lists of file names  ===  ===

      .OPENR  #2  'MACLST'
      .IF <FILERR> <> 1 .GOTO FE0020
      .READ  #2  JUNK          ! throw away title line
      .IFT <EOF> .GOTO NDMACL
      .IF <FILERR> <> 1 .GOTO FE0020

      .OPEN  #3  'MACTMP'
      .IF <FILERR> <> 1 .GOTO FE0030

.;  ===  ===  Loop through list of names  ===  ===

```

```

.;      = = =  Parse next name  = = =

.DOMACL:
    .READ #2 FNAME          ! get next line from file
    .IF <FILERR> <> 1 .GOTO FE0020 ! quit - no more
    .TEST  FNAME           ! check for final line
    .IF <STRLEN> = 2 .GOTO NDMACL ! final line - just ^M^J
    .SETS  S1A      FNAME[3:*] ! take the name
    .PARSE S1A      "." S1 NULL ! drop the extension

.;      = = =  Include name in macro command file  = = =

    .SETS  INSPEC  DEV+SRCU+S1
    .SETS  LSTFI   S1+";1"
    .SETS  OBJFI   DEV+SRCU+S1+";1"
    .IFF   LSTS    .DATA #3 'OBJFI''OSW'='INSPEC'
    .IFT   LSTS    .DATA #3 'OBJFI''OSW','LSTFI''LSW'='INSPEC'
    .IF <FILERR> <> 1 .GOTO FE0030

.;      = = =  Include name in library create file  = = =

    .DATA #0 'OLBFI'/IN='OBJFI'
    .IF <FILERR> <> 1 .GOTO FE0000

.;      = = =  Include name in object module cleanup file  = = =

    .DATA #1 'DEV''SRCU''S1'.OBJ;1/DE
    .IF <FILERR> <> 1 .GOTO FE0010

.;      = = =  Done processing name  = = =

    .GOTO DOMACL

.;      === ===  Done processing all names in file  === ===

.NDMACL:
    .CLOSE #2
    .CLOSE #3

.; * * * * *
.; *  Terminate command file creation  *
.; * * * * *

    .CLOSE #0          ! close libr, pip cmd files
    .CLOSE #1

.; *****
.; *  Build the library  *
.; *****

.; *** ***  do the assembly  *** ***

```

```

MAC @'MACTMP'

.; *** *** create object module library *** ***

LBR @'LBRTMP'

.; *** *** clean up object module library *** ***

PIP 'OLBFI'/TR

.; *** *** clean up objects *** ***

PIP @'CLNTMP'

.; *****
.; * Terminate processing - return to starting uic *
.; *****

.EXITCF:
PIP 'LBRTMP'/NM/DE, 'CLNTMP'
PIP 'MACLST'/NM/DE, 'MACTMP'

.EXIT <SUCCES>

.; * * * * *
.; * File i/o error handling *
.; * * * * *

.FE0000:      .; * * * Error writing to LBR command file * * *

; ??? I/O error (code '<FILERR>') writing to temp file 'LBRTMP'
.GOTO FE9999          ! terminate processing

.FE0010:      .; * * * Error writing to PIP command file * * *

; ??? I/O error (code '<FILERR>') writing to temp file 'CLNTMP'
.GOTO FE9999          ! terminate processing

.FE0020:      .; * * * Error reading PIP-created file * * *

; ??? I/O error (code '<FILERR>') reading temp file 'MACLST'
.GOTO FE9999          ! terminate processing

.FE0030:      .; * * * Error writing to .MAC command file * * *

; ??? I/O error (code '<FILERR>') creating temp file 'MACTMP'
.GOTO FE9999          ! terminate processing

.FE9999:      .; * * * Common error termination
.CLOSE #0
.CLOSE #1

```

```
.CLOSE #2  
.CLOSE #3  
.EXIT <ERROR>    ! terminate processing
```

```
.ENABLE GLOBAL
.ENABLE SUBSTITUTION
```

```
.;+++++
.;
.; GRPCOP.CMD - Copy a group's worth of files to a different uic
.;
.; Command file copies of a group's worth of software to a different
.; group. Invoke command file one of two ways:
.;
.;     either @GRPCOP sdevgp tdevgp
.;     or     @GRPCOP +[sw] sdevgp tdevgp
.;
.; where:
.;     sdevgp is source file device plus ":" plus group
.;     tdevgp is source file device plus ":" plus group
.;     "sw" is any combination of the switches below:
.;         "D" - forces deletion of the sgroup files and uics after copy
.;         "R" - use PIP "rename" rather than "copy" command to move files
.;
.; NOTE:
.;     Command file checks for the following parameter errors -
.;         null device, or null group
.;         non-octal group
.;         invalid device
.;         same physical devices, same group
.;         rename requested using different source, target devices
.;     Command file does not check for the following errors -
.;         device driver not loaded
.;         device offline, not mounted, mounted foreign, or
.;         attached/ allocated to another task
.;
.; Command file returns <SUCCES> or <ERROR> status depending on whether
.; the copy completed ok.
.;
.;-----
```

```
.; *****
.; * Preliminary processing - determine command file parameters *
.; *****
```

```
.; *** ** Decode caller switches *** **
```

```
.SETF DELE
.SETF RNAM
.SETS SRCSPC P1          ! assume no switches at first
.SETS TGTSPC P2
.TEST P1 "+"
.IF <STRLEN> <> 1 .GOTO SWDONE
.SETS SRCSPC P2          ! switches present
.SETS TGTSPC P3
```

```

.TEST P1 "D"
.IF <STRLEN> <> 0 .SETT DELE
.TEST P1 "R"
.IF <STRLEN> <> 0 .SETT RNAM      ! rename supersedes delete
.SWDONE:

.; *** ***   decode and test device/group specifications      *** ***

.; First, parse out the specifications

        .PARSE  SRCSPC  ":"  SRCDEV  SRCGRP
        .SETS   SRCDEV  SRCDEV+":"

        .PARSE  TGTSPC  ":"  TGTDEV  TGTGRP
        .SETS   TGTDEV  TGTDEV+":"

.; Second, do some semantic checks on the specs

.; * * * *   Source device   * * * *

        .TEST   SRCDEV
        .IF <STRLEN> <> 0 .GOTO SDVNNL      ! source device nonnull
        ;
        ; ??? Error - no source device specified
        ;
        .EXIT <ERROR>
.SDVNNL:
        .TESTDEVICE 'SRCDEV'
        .SETS SRCSTR <EXSTR1>             ! get device info back
        .TEST SRCSTR "NSD"                ! does device exist?
        .IF <STRLEN> = 0 .GOTO SDVEXS     ! yes, in config
        ;
        ; ??? Error - non-existent source device ('SRCDEV')
        ;
        .EXIT <ERROR>
.SDVEXS:
        .PARSE  SRCSTR  ","  PSCDEV  JUNK      ! get physical devnam

.; * * * *   Target device   * * * *

        .TEST   TGTDEV
        .IF <STRLEN> <> 0 .GOTO TDVNNL      ! source device nonnull
        ;
        ; ??? Error - no target device specified
        ;
        .EXIT <ERROR>
.TDVNNL:
        .TESTDEVICE 'TGTDEV'
        .SETS TGTSTR <EXSTR1>             ! get device info back
        .TEST TGTSTR "NSD"                ! does device exist?
        .IF <STRLEN> = 0 .GOTO TDVEXS     ! yes, in config
        ;

```

```

; ??? Error - non-existent target device ('TGTDEV')
;
.EXIT <ERROR>
.TDVEXS:
.PARSE TGTSTR "," PTGDEV JUNK ! get physical devnam

.; * * * * Source group * * * *

.TEST SRCGRP
.IF <STRLEN> <> 0 .GOTO SGPNNL ! source group nonnull
;
; ??? Error - no source group specified
;
.EXIT <ERROR>
.SGPNNL:
.IFT <OCTAL> .GOTO SGPOCT ! group must be octal
;
; ??? Error - source group ('SRCGRP') not octal value
;
.EXIT <ERROR>
.SGPOCT:
.SETN SG 'SRCGRP'
.IF SG <= 377 .IF SG > 0 .GOTO SGPINR ! group in range
;
; ??? Error - source group ('SRCGRP') not in 1-377 range
;
.EXIT <ERROR>
.SGPINR:

.; * * * * target group * * * *

.TEST TGTGRP
.IF <STRLEN> <> 0 .GOTO TGPNNL ! target group nonnull
;
; ??? Error - no target group specified
;
.EXIT <ERROR>
.TGPNNL:
.IFT <OCTAL> .GOTO TGPOCT ! group must be octal
;
; ??? Error - target group ('TGTGRP') not octal value
;
.EXIT <ERROR>
.TGPOCT:
.SETN TG 'TGTGRP'
.IF TG <= 377 .IF TG > 0 .GOTO TGPINR ! group in range
;
; ??? Error - target group ('TGTGRP') not in 1-377 range
;
.EXIT <ERROR>
.TGPINR:

```

```

.; * * * * source vs. target device/group

.IF PTGDEV <> PSCDEV .OR .IF TGTGRP <> SRCGRP .GOTO SPCSDF
;
; ??? Error - physical source, target group/devices
; are identical ('PSCDEV', group 'SRCGRP')
;
.EXIT <ERROR>
.SPCSDF:
.IFF RNAM .GOTO RNAMOK
.IFT RNAM .IF PTGDEV = PSCDEV .GOTO RNAMOK
;
; ??? Error - cannot rename between different
; physical source, target devs ('PSCDEV', 'PTGDEV')
;
.EXIT <ERROR>
.RNAMOK:

.; *** *** format source, target group designations *** ***

.SETS SRCGRP "'SG%O'"
.IF SG < 10 .SETS SRCGRP "00"+SRCGRP
.IF SG >= 10 .IF SG < 100 .SETS SRCGRP "0"+SRCGRP

.SETS TGTGRP "'TG%O'"
.IF TG < 10 .SETS TGTGRP "00"+TGTGRP
.IF TG >= 10 .IF TG < 100 .SETS TGTGRP "0"+TGTGRP

.; *** *** set up synonyms *** ***

.SETS PIPTMP "PIP.TMP;1" ! *** PIP temporary name list
.SETS TEMPLUN "2" ! *** lun for list read

.; *****
.; * Generate list of source group uics *
.; *****

PIP 'PIPTMP'='SRCDEV'[0,0]'SRCGRP'%%.DIR/LI
.LSTGOT:

.; *****
.; * Do the uic by uic copy *
.; *****

.; *** *** Init - open uic list for current group *** ***

.OPENR #'TEMPLUN' 'PIPTMP'
.IF <FILERR> = 1 .GOTO PTOPOK
;
; ??? Temp file open error: 'PIPTMP' (code '<FILERR>')
;
.EXIT <ERROR>

```

```

.PTOPOK:
    .READ    #'TMPLUN' JUNK          ! throw away title line

.;    *** ***  Loop through directory list      *** ***

    .;    * * *  Parse out next member field, if spec present * * *

.NXTUIC:
    .READ    #'TMPLUN' ULINE
    .READ    #'TMPLUN' JUNK
    .IFT <EOF> .GOTO ALLCOP
    .TEST    ULINE SRCGRP
    .SETS    UMEM ULINE[<STRLEN>+3:<STRLEN>+5]

    .;    * * *  Define synonyms for source, target uics * * *

        .SETS    SUIC    "[ 'SRCGRP', 'UMEM' ]"
        .SETS    TUIC    "[ 'TGTGRP', 'UMEM' ]"

    .;    * * *  Transfer files * * *

        ; Transferring member 'UMEM'
        .ENABLE QUIET
        .; create new directory on target device if needed
        .TESTFILE 'PTGDEV'[0,0]'TGTGRP''UMEM'.DIR
        .IF <FILERR> <> 1 UFD 'PTGDEV''TUIC'
        .; transfer files
        .IFF RNAM PIP 'TGTDEV''TUIC'/CD='SRCDEV''SUIC'
        .IFT RNAM PIP 'TGTDEV''TUIC'/RE='SRCDEV''SUIC'
        .; set files to proper access code
        PIP 'TGTDEV''TUIC'*.*;*/PR/FO

    .;    * * *  Delete sources, if desired * * *

        .IFF RNAM .IFT DELE PIP 'SRCDEV''SUIC'*.*;*/DE
        .IFT DELE PIP [0,0]'SRCGRP''UMEM'.DIR;1/DE
        .DISABLE QUIET

    .;    * * *  Continue * * *

        .GOTO    NXTUIC

.ALLCOP:

    .;    * * *  End of loop to copy one uic * * *

.;    *** ***  End procedure      *** ***

    .CLOSE #'TMPLUN'
    PIP 'PIPTMP'/DE

    .EXIT <SUCCES>

```



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

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

**MOVING OR REPLACING A DELEGATE?**

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

( ) Change of Address  
( ) Delegate Replacement

DECUS Membership No.: \_\_\_\_\_

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

State/Country: \_\_\_\_\_

Zip/Postal Code: \_\_\_\_\_

Phone No.: \_\_\_\_\_

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