# THE MULTI-TASKER

Volume 16, Number 1 July 1982

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

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

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

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

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

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

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

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

## TABLE OF CONTENTS

### Columns

### Articles

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.

## READ THIS FIRST

A critical problem to BRU is discussed in this issue. Carl Mickelson did an impressive analysis of a backup problem and uncovered a serious flaw in BRU's sort algorithm. It also appears that this problem could be related to the incompatibility between V3.2 BRU and V4.0 BRU.

One of the largest working group's in the SIG is Process Control. They are collecting data about themselves and a questionnaire is found in this issue. If you have any interest at all in real-time control, please answer the questions and return to the Multi-Tasker. We will forward the responses onto the working group.

Finally, on the next page is a note from Steve Paavola on the status of RSX-11M. This should clear the air about any rumors you may have heard about the "death" of RSX-11M.

## From the Editor

You will want to read and reread this issue from cover to cover. To start with, the DECUS/RSX SIG Library News column carries preliminary information on the submissions to the Atlanta RSX SIG Tape. For articles, you have two submissions related to RSX-11M and VAX/VMS and one on RT-11 to RSX-11M conversion. Technical articles cover how to implement home directories in RSX-11M and the KMS Fusion enhancements for RSX-11M V4.0.

Next month, I will try to publish several articles which are long overdue. I am in the process of tabulating the SPR Survey and will have a analysis of the results next month. Also, I will sort through the pages of notes from the Atlanta Symposium and report on what happened. Hopefully this will include transcripts of the various question and answer sessions and product panels.

Finally, the Multi-Tasker has received over 500 SPR's in various form as a part of the SPR Survey effort. I will be working with the Retired Versions of RSX-11M working group to sort through them and publish the most critical ones.

Ralph Stamerjohn
Multi-Tasker Editor
June 9, 1982

Phone: (314) 694-4252 (3-5 pm, CST)

# The Mellowing of RSX-11M

Steve Paavola
RSX Product Manager

At DECUS in Atlanta, I announced that the RSX-11M operating system is approaching the maximum level of functionality planned for it. In detail, its status is:

- Version 4.0 has been released. We believe it contains about as much functionality as it makes sense to support on systems with less than 256 KB. RSX-11M will continue to receive bug fixes, new device support, and maintenance releases. But major new functionality, like DCL in Version 4.0, should not be expected.

- Layered product enhancements will continue to occur. In general, it makes more sense to put new versions of products on RSX-11M as well as RSX-11M-Plus, rather than doubling our support effort.

- This support will continue for the indefinite future. The vast majority of our current RSX-11M users either cannot upgrade to RSX-11M-Plus because of hardware limitations, or choose not to for a variety of reasons. We have thousands of maintenance contracts with these customers, and they are buying significant amounts of hardware, so we have no intention of neglecting them.

- Those RSX-11M customers that do meet the minimum hardware requirements, should look seriously at upgrading to RSX-11M-Plus. RSX-11M-Plus supports all of our current PDP-11 processors and will be receiving enhancements in the future.

  We are working hard to make RSX-11M to RSX-11M-Plus migration easy. Tasks, except privileged tasks that access the executive data base, migrate without conversion. User-written drivers require some conversion, but this is usually easy and we have documentation on how to do it now.

  Once the customer makes it to RSX-11M-Plus, he has a higher performance, higher functionality system that he started with. Plus, he has the added growth potential of RSX-11M-Plus.

In summary, RSX-11M is the second best 16-bit multi-tasking real-time operating system in the industry, second only to RSX-11M-Plus. We intend to exploit that capability.

# From Five Years Ago

Gail Green
Multi-Tasker Historian

The July 1977 issue of the Multi-Tasker (Vol. 8, No. 1) reported on events of the Spring '77 symposium. The SIG bylaws were adopted. The bylaws were published in this issue of the Multi-Tasker. Digital responded to the resolutions adopted at the Fall '76 symposium in Las Vegas. The resolutions resulted in:

1. The MOP Fortran compiler and OTS sources becoming available through the Software Distribution Center.

2. The announcement of a Digital project to provide machine readable "bug fix" update packages (autopatch kits), separate and distinct from new releases.

3. A Digital reply that the RSX-11D Hotline would not be expanded to include calls on IAS, RSX-11S, and RSX-11M. The Hotline was implemented as a remedial effort for versions 6b and 6.2.

This issue also included some bug fixes for RSX-11M V3.0 and details on how to implement multiple device de-spooling by using multiple copies of the printer spooler.

# DECUS/RSX SIG Library News

Paul Tompkins
Library News Editor

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

## New DECUS C Release

Bob Denny has submitted, to the Spring 1982 RSX SIG Tape, a "PRELIMINARY binary kit for Decus C" which supports in-line floating values and EIS instructions. The final release kit (complete with all sources) will be available from the DECUS library soon.

It is unclear when the new source kit will be available. It will not be until at least August 1982 and possible several months later. The Multi-Tasker (and other DECUS publications) will carry an announcement when the new version becomes available.

If you want to get started with C, the DECUS office will continue to process orders for the old version. While you will have to re-order to get the new version, given the low cost for what you are getting, this is still one of the best bargains you will ever find.

## Preliminary Spring 1982 RSX SIG Tape Abstracts

The following is a collection of the README files on the Spring 1982 RSX SIG Tape from Atlanta. The text has mostly been editted for formatting purposes only; the content is from the actual submiters. The files submitted must be checked for possible copyright problems and the final distributed tape may not have everything in this list. Also, the final master distribution may change account numbers to match the standards established for the SIG tapes.

The distribution of the Atlanta tape will again use the SIG tape tree. Past experience shows that at least six months are needed for the tapes to move through the entire tree, so it could be until November before the tape becomes available in your area. When the tree is assembled, it will be published in the Multi-Tasker.

* [1,121]

  This UFD contains three submissions from the Southest Minnesota LUG.

  - INSPS.SLP is an untested (!!) SLP correction file for RSX V4 INStall task, and implements the INS syntax '/TASK=...xxx /CMD="xxx command line"' in a single switch, /RTC. The new switch syntax is: '/RTC="xxx command line"', and saves 13 bytes for passing MCR command lines via CCL or whatever.

  - DIR was originally a modified SRD V5.0 implemented to use a more "DCL" style command syntax. It has been enhanced over 2 years and includes several options not (to my knowledge) supported by SRD: 1) handles PURGE operations, 2) may select null length and locked files, 3) may reset file creation/revision dates, 4) may set, delete, and list files based on an EXPIRATION date (taking advantage of unimplemented exp. date section of FILES-11 file header). DIR.RNO contains a RUNOFF compatible documentation set.

  - HP67 is a Hewlett-Packard programmable calculator emulator which currently runs under RSX-11M and RT-11. It is written entirely in FORTRAN; includes 256 double precision "registers" and 1000 program steps. HP67.DOC contains a full description, COMMAN.DOC contains a list of valid instructions.

* [7,77]

The program STARFLEET was written by Trevor Sorensen for use on the RSX-11M(+) Operating System. STARFLEET is a fairly standard Star Trek-type program, although more sophisticated than most. However, one feature that sets it apart from most Star Trek games is the incorporation of a comprehensive Service Record and associated features. The program is written in FORTRAN IV-PLUS, and is set up to be installed on UIC [200,200].

* [260,22]

This account has various documentation files, which will also be published in future issues of the Multi-Tasker. Topics include:

- Description of a way to implement safe INStall/REMove for non-privileged users of larger RSX11M system.

- Example of useful enhancement of the well-known POOL monitor task.

- How to significantly increase throughput of an overloaded RSX11M system. SHUFL.COR SLP file for SHUFFLER included.

- Super-version of CATCH-ALL system. No modification to standard executive, but works the same way if patch to DRSPW realized. Many other enhancements. All necessary files included.

* [300,121]

This UIC contains a series of Fortran callable subroutines designed to allow the user to utilize the internal characteristics of the VT-100 family of terminals. Since the original purpose of this library was to support a VT-105 terminal, some of the routines are specific for that type terminal. These routines were written for RSX-11M (VERS 3.2), and the full duplex terminal driver.

* [301,66]

This is a modified version of RUNOFF (M2.3) from Bradford A. Lubell of the Los Angeles Cardiovascular Research Laboratory. The following modules will allow one to build RUNOFF (M2.4) for the operating systems: RT-11, RSX-11, RSTS/E, IAS, VMS. Features added will only support Diablo (and Diablo compatible) printers and the Anderson-Jacobson 832 terminal. These features can be switched off at runtime. The new features are:

- The pitch is changed instead of adding spaces to fill-out the right margin.

- Control over the base horizontal pitch and the vertical pitch.

o Superscripting and subscripting.

o Ribbon color select (Diablo only).

o Print enhancement mode (Anderson-Jacobson only).

o Alternate character set selection via shift-in/shift-out character transmission (any terminal).

* [301,65]

The REF program in conjunction with the TREE program will perform functions such as output cross-reference data, output user and system symbols, output the calling tree structure, and output an automatic overlay description file for your program. The .ODL file will have to be edited to conform to your system configuration with respect to resident libraries and such. The sources are given in OMSI PASCAL version 1 and MACRO-11. All necessary documentation for running the REF program can be found in the REF.CMD file.

* [307,30] to [307,36]

This is the latest release from Joe Sventek of the Lawrence Berkeley Laboratory Software Tools Virtual Operating System. The release is contained in seven accounts. Note that account [307,36] is the first release of the Software Tools network-wide mail system.

o [307,30]TOOLGEN.CMD: the command file for building the LBL Software Tools Virtual Operating System. For more information on the features available in this release, consult the release notes contained in this UIC, RELEASE.DOC.

o [307,31]: Everything you need to build this release of the VOS.

o [307,32]: The source files for the RSX-specific tools and primitives.

o [307,33]: The source files for the portable VOS utilities.

o [307,34]: Source files for the variable-length send/receive driver. In addition, a document describing home directory management for RSX may be found here.

o [307,35]: Archives containing the format input files for the manual entries in sections 2-4 of the manual.

o [307,36]: The initial distribution of the Software Tools Mail System.

* [307,50]

Here is yet another version of RUNOFF! This version is the result of an effort to achieve the ability to produce an extensive customer system manual without having to do any manual editing of the RUNOFF output. Toward that end, several new commands have been added, several existing commands have been enhanced, and many bugs have been fixed. This version is derived from the current "official" DECUS version, M02. This version has been used only under RSX-11M. I have no idea whether it will still run correctly under RSTS or IAS -- presumably it will. The user's manual has been edited extensively to correct errors, add clarity (hopefully), and document changes to the program. (Option "switches" within the RNO file select inclusion or exclusion of directions for working under RSX or RSTS.) Significant feature enhancements in this version include:

o The task now expands itself when the index/footnote buffer fills. Thus, RNO now does not occupy any more memory than it actually needs.

o An assortment of functions have been added for making the output alternate left/right for reproduction on both sides of the page.

o Multiple input files can be specified.

o Points at which a word can be hyphenated can be explicitly specified (to overrule RUNOFF's error-prone algorithm).

o The index can include subentries, as well as entries without page references (for headings above subentries).

o A base left margin can be set from which the text margins are measured.

o Command syntax is now the same for all commands with similar argument combinations.

o Sections of the input can be set off with command-line control over whether or not they are to be included in the output.

* [307,110]

This account is from the California Regional Office of the U.S. Forest Service, Information System Management staff. This account is a combination of some useful RSX-11M V3.2 DECUS stuff (some modified by us for local usage), most of which has been tested and/or run under RSX-11M V4.0, and some U.S. Forest Service routines, all tested and run under RSX11M V4.0.

* [311,1]

This account has a DN11 Device Driver for RSX-11M. The supplied tape for this driver contains the sources and indirect file for assembly prior to inclusion into the system. Follow the procedures outlined in the DN.CMD file for inclusion of this driver into your system. The DN11 device driver was developed to support auto-call procedures under RSX-11M .The driver is implemented as a user written driver, non-loadable. It supports only one physical device, with a maximum of four output ports.

* [312,12]

This account has an updated OPA (On-line Pool Analyzer) for RSX-11M V3.2. OPA displays a visual map of the data structures in pool and their location, thus allowing system programmers/managers to determine why their pool is so fragmented, and hopefully then rectify the situation. This is a snapshot of a running system, not a crash dump tool.

* [312,14]

A very simple program for undeletion/reincarnation of programs.

* [312,15]

A STAGE2 macroprocessor, complete with its test data, and a copy of SIMCMP used in its implementation. REF: Waite. Implementing software for Non-Numerical Applications.

* [312,17]

This account has an updated version of the device database lister on the San Diego 1980 SIG Tape (account [301,31]]. The utility displays device (driver) database (i.e. DCB, UCB, SCB) addresses, contents, and offset labels. Modified for M+ and symbolic displays of BIT masks, and also to detect ".-2" as a special case.

* [312,20]

This account has corrections to the Spring 1981 TH and AT RMDEMO pages, and a quick hack of a page called DA. DA is a page to display the device activity, It runs up to a rate of once every 3 ticks, and displays the list of io packets waiting for, or the packets active on, each major device in the system. Also displayed is the list of tasks currently waiting for the processor.

* [312,315]

This account has various utilities from Richard Kirkman. Most notable are M+ versions of OPA and the BRUDIR program which gives, in ONE PASS, a very complete directory of a BRU tape.

There is also Glenn Everhart's latest stuff, including:

o DDT22: A new release. Fixes some bugs, adds macros so you can automatically execute a set of commands at a breakpoint. The new version also supports the I and D space option of RSX11M+ V2.

o DGTUNX: A general tape mover, read/write fixed record length files, and has new ability to read UNIX TAR format tapes. Note: the block factor may have to be changed to 1 for this to work reliably. BIGTPC bouncing an image mode copy off a virtual disk will do it...

o BIGTPC: My latest version, has image mode copy support for moving around images of virtual disks. Also can copy disk to disk in container files. Can create pseudo-FLX tapes with disk images with block factors up to 72.

* [312,330]

A sort package from Glen Hoffing. Looks like the DEC SORT-11 but uses FCS, not RMS. [312,320] is an earlier version.

* [312,322]

Glenn Everhart's latest VE: driver. This is a revision to VD: that will work on M or M+ (V3.2 or V4 of M, or V1 or V2 of M+) which can be built to allow multiple files or devices or mixtures of them to logically be combined onto a single disk. Current source supports up to 8 files or device areas (by filename or by LBN and length) per virtual disk. The name is changed to VE: to permit it to coexist with the older VD: type disks. This allows you to have NONCONTIGUOUS VIRTUAL DISKS, or to pretend that 5 or 6 of your old RL01's and RK05's are really a big disk. Biggest contiguous space will still be the size of the real largest contiguous space (the driver invents some phony "bad" blocks), but your total volume will be able to be large. [312,327] is an extra copy.

* [312,331]

A PACMAN by Glen Hoffing. This game works on VT100's and does NOT require advanced video. Looks very like the arcade game PACMAN. Control C erases the screen. (It works OK on VAX too!) Several versions of the source are on the tape; the highest UIC is the latest and presumably best debugged. Earlier versions will allow higher scores supposedly and are supplied for the ego gratification of those who want to amaze their friends by appearing good at the game. Uses the 4,8,6, and 2 keys for controlling PACMAN's direction. The game even includes the fruits. Source is in RATFOR which may make it easy to convert to C. [312,321] and [312,325] are earlier versions.

* [312,332]

This is an improved version of SRD, dated 3-DEC-1980 from me (Phil Stephensen-Payne). This version is intended to combine all the versions of SRD I know of (Ray van Tassle's, Bob Turkelson's and my own) into one (easily tailorable) unit, which can then hopefully be used as a base for future modifications. For full details on this version of SRD consult SRD.DOC.

* [352,1]

This UIC contains correction files for SRD V6.0 by me (Bob Turkelson), which was submitted to the Spring 1981 Miami RSX SIG tape, in [373,4] The modifications I made are:

- o Corrected the problem with SRD.CMD which caused the generation of incorrect initialization defaults when the <ESCAPE> feature was used.

- o Changed SRD.CMD so that a different SRD.ODL file is generated when SRD is built with FCSRES.

- o Corrected a bug which prevented /WB/-LI from working properly (the directory was not sorted before the write-back).

- o Changed SRD so that when the /SE switch is specified, /WI and /WD are ignored; that is, there is no implicit wild-carding whenever the /SE switch is used.

- o Added the /ZE switch to list zero used block files.

- o Added code so that if the file attributes area of the header contains a value of 0 for the number of blocks allocated, SRD will calculate the number of allocated blocks from the retrieval pointers.

* [315,200]

This program is from Phil Rowland and will allow the user to create, read and/or write IBM Basic Exchange diskettes. It has been tested on RX01's only. The user can:

- o Initialize diskettes.

- o Transfer Files 11 files to Data Sets.

- o Transfer Data Sets to a Files 11 files.

- o Perform ASCII/EBCDIC or EBCDIC/ASCII code conversions.

- o List Data Set Labels (directories).

- o Delete Labels.

* [326,112]

This account has SPELL,a spelling checker from Jeff Hamilton of E-Systems, Greenville Division.

* [326,113]

This account has submissions from Mostek. All of the following utilities run under a RSX V3.2 (e) system with a 128k PDP-11/34/60 as the CPU. VAX programs submitted will run under VMS 2.4 and later releases with PHASE III (VAX V2.0) DECNET.

- o CLK: An extremely simple switch register time-of-day clock which displays hours, minutes, and seconds in octal on the console switch register. This task runs on a PDP-11/60 with a 4k IOPAGE device register partition.

- o JOB: This utility spawns a copy of the indirect on a specified terminal with a file of a specified name. This may be useful in non-mup systems in which you may wish to initiate captive indirect command procedures on remote terminals within the system STARTUP.CMD startup command procedure.

- o SRDCMD: This indirect command procedure is used to sort an entire disk volume directory structure. Used from a privileged terminal, it can sort and compress a disk directory without user intervention. Note: check to insure that you version of SRD generates directory listings which correspond to the format required by this command procedure before using it!

- o RJEMSG: This network server task is designed to receive SDAT$ data packets and transmit them to remote VAX users who are currently linked via the EXtended Remote Batch (XRB) facility. In this fashion, remote VAX users may transmit or receive command lines to properly designed RSX tasks.

- o XRB: This is the VAX partner to RJEMSG. This piece of VAX FORTRAN establishes a logical link between the VAX process and the remote RJEMSG task and allows the user to send/receive messages to the remote network server.

- o RMS: Support routines for XRB.FOR

- o STRING: Support routines for XRB.FOR

- o TNM: A simple utility designed to enable anyone to change a task's name from one RAD50 character string to another RAD50 character string.

- o TRD: A magnetic Tape ReaD utility which reads ASCII/ binary mag tape data and appends it with ASCII/ binary disk files to produce EBCDIC disk files suitable for binary transmission over an RJE/HASP communications link.

- o TWT: This is the counterpart to TRD, a Tape WriTe utility. As binary punch files return from a remote host and are stored on disk, this utility may be invoked to take the 80-byte binary punch records and format them to mag tape.

* [330,1] to [330,17]

This tape contains release 3 of "The Best of ICR", a selection of useful programs from the Institute For Cancer Research (Bob Stodola, Bill Cael, Bill Wood). Almost all of the programs have been revised in one way or another since the last release to fix bugs or add capability. Following are some random notes:

1. Many of the programs use a macro interface to the FCS file system. If your installation uses the RMS file system by default, you will have to link these programs with the FCS library.

2. The version of this tape which was distributed on the Fall 1981 RSX/IAS SIG tape had a bug which caused LIST, SRDCMD, BURSTF, and RATFIV not to come up properly on RSX systems, ie. these programs attempted to open "TO:", which is not a legal device on RSX systems. This bug has been fixed, and LIST and RATFIV have been updated considerably.

3. The bug in the RSX subtasking subroutine, RUNSYSRSX, which was distributed on the Spring 1981 RSX/IAS SIG tape has been fixed. This means that programs that do subtasking such as CPL, SRDCMD, and TRU should work now on RSX systems.

4. Most of the programs may be linked with a resident library to make the tasks smaller.

5. When building the programs, be sure to read each one into its own UIC to avoid file name conflicts (for instance RATFIV's IO.RAT is different than LIST's IO.RAT).

6. When building FDT on RSX systems, be sure to uncomment the line in FDT.MAC which defines RSX11M.

7. On IAS, programs which do subtasking (TRU, CPL, SRDCMD, and LIST when using SRD to select files) require subtasking privileges, and XEQ requires the chain privilege.

8. CPL will run on RSX or IAS systems, contrary to rumors that it is an IAS only program.

9. SRD has been substantially debugged since the last release and has none of the horrible SRD bugs that some SRDs have, as mentioned on SIG tapes and at the Fall 1981 DECUS convention. On one RSX system, SRD had to be compiled with BIGMAC, as regular MAC failed.

10. BURSTF now handles files with characters in columns 73-80.

A short description of each program on the tape follows:

o [330,1]LIST: LIST is like the TYPE and GREP commands only better. LIST's simplest use is to display lines from a file a screen at a time, however it has sophisticated string searching, can be positioned randomly in the file, and can write portions of the file to other files. In addition, LIST can use SRD to select multiple files, making it easy to do such chores as searching through a group of files for a particular string or variable. RSX or IAS.

o [330,2]FDT: Number one on the RSX and IAS menus was a Fortan symbolic debugger. Here it is! RSX or IAS.

o [330,3]CPL: CPL compiles modules in any language if the source has been changed since the last compilation. It handles multi-pass compilers, including RATFIV, Swedish and NBS Pascal, and Whitesmith's C, as well as single-pass compilers such as F4P, BASIC+2, and MACRO. A must for software developers. RSX or IAS.

o [330,4]SRDCMD: This is a flexible command line generator which spawns SRD to select files, then writes command lines con- structed from specified text and the device, uic, name, exten- sion, and version fields of the selected file names. RSX or IAS.

o [330,5]SRD: An enhanced version of SRD including more powerfull file string matching, selection by file revision date, and storage allocation summary. RSX or IAS.

o [330,6]TAPE: TAPE reads and writes tapes in a variety of formats, including blocked ASCII, blocked EBCDIC, and DEC-10 copy format. RSX or IAS.

o [330,10]BURSTF: BURSTF bursts Fortran subroutines, functions, main programs, and block datas from a file and writes them to individual files. RSX or IAS.

o [330,11]RESEQ: This utility resequences the statement labels in a Fortran program. RSX or IAS.

o [330,12]TRU: TRU truncates files, but does not access the file if it doesn't need truncation. This is a must when using BRU for incremental backup, as BRU would copy all files which had been truncated with PIP, even if the file hadn't needed truncation. RSX or IAS.

o [330,13]XEQ: XEQ runs programs. It maintains a global common area which contains, for each user, a list of directories to search when XEQ trys to run a program. Never again will you have to type XX0:[100,222] when running XX0:[100,222]FOO! In addition, XEQ can pass a command line to the program it invokes, whether or not the program is installed. IAS only.

o [330,14]SCHEDULE: SCHEDULE will invoke MCR command lines at scheduled times during the week. The list of commands and times is maintained in a file. IAS but easily modified for RSX.

o [330,14]GAME: GAME uses SCHEDULE to restrict game playing to specified times. IAS and RSX (see SCHEDULE).

- [330,14]RUNNL: This is a companion program to SCHEDULE and GAME. It will run a task on the pseudo device NL. IAS only.

- [330,15]RATFIV Version 2: An enhanced and debugged version of the Software Tools Group's Ratfor compiler, Ratfiv features optional Fortran 77 output, the SWITCH and STRING statements, a powerful macro processor with macro argument capability, conditional compilation, output of string literals as quoted strings (necessary for OPEN statements, etc.) or optionally as hollerith strings, and specification of FORMATs within READ, WRITE, ENCODE, and DECODE statements. RSX or IAS.

- [330,16]MTREK: The best Star Trek ever, this is a multi-terminal real-time all out war game, replete with torpedoes, tractor beams, anti-matter, etc. This program was originally written at Boeing; we have enhanced and debugged it. Terminals with clear screen and cursor positioning are all that's required. IAS or VAX, but you could modify it for RSX.

* [332,100]

This account has two submissions from Bill Burton. First is the final version of two programs for doing partial restores of disks from DSC tapes. These programs first appeared on the Spring 1981 RSX tape.

Second is a reprint and patch files from the MARCH, 1982 issue of the Multi-Tasker. These files patch the FLX utility distributed with rsx-11m versions 3.0 and 3.1 to allow 9-character file name catalogs and searching.

* [332,110]

This account has TAB from R.N. Stillwell. This is a low-overhead data management system. These programs provide a small, relatively unsophisticated data base management system for the PDP11 in which the user can easily define and manipulate his own data tables without requiring the assistance of a programmer. Facilities provided by this system are:

- To define a table, or modify the format of an existing table.

- To edit a table, correcting data and entering new data.

- To list an entire table or selected parts of a table.

- To generate a printable report file including standard text, values from one or more tables, and user-entered or computed values; to update tables, invoke subsidiary tasks, and perform general arithmetic and character-string operations.

- To extract a subtable consisting of a set of contiguous rows from a table.

- To merge two tables having the same number of columns and the same lengths of each column.

- To convert a table to a sequential file for transfer to a different system, and to restore the table from the sequential file.

All source files required to build the tables programs are provided; languages used are Macro and Flecs. An object library and command files to build the Flecs preprocessor are also provided. Both IAS and RSX-11M versions are provided.

* [333,1]

This account has various submissions from the Washington DC LUG, prepared by Don Rubin. Included are the following:

- EDTINI: our initialization file for EDT, with some good key definitions.

- EXE: a command file to F77 compile and build programs.

- IBMFIL: a command file used to compile and build programs with input from the keyboard or a file.

- LIFE: the game of life modified for VT100 terminals.

- TIMLOG: Programs used with the KMS accounting system to implement chargeback accounting on an 11M system. Please refer to ACC.RNO before fooling with these.

- QS: More programs to foll with the files generated by the KMS accounting system.

- Various documents on RSX system notes and documentation on how to do a system generation.

* [333,100]

This UIC has submissions from Allan Watson. It contains one super program, a few useful or interesting command files and TECO macros, and a couple of simple programs in Pascal (Swedish compiler from DECUS) and "C" (also from DECUS).

- DSUTIL: Data Set Utility which allows a priveleged user to dynamically examine data from devices or files on a block-by-block basis. DSUTIL allows the user with a VT100 to display the data in various formats: octal word, octal byte, hexadecimal word, RADIX 50, ASCII, or EBCDIC. Logical or virtual blocks can be accessed randomly or sequentially. Access can be by device and block number or by file name. DSUTIL also displays UFD's. Display mode can be changed from one to another without re-reading the block.

- WHATS: Macro program that, when RUN, clears a VT100 screen and then displays the octal value of any key you strike, including control sequences.

- TABLEGEN.TEC and MACTOCCL.TEC: These TECO macros may be used in conjunction with Jim Downward's CCL to convert a .CCL file to MACRO-11 format for inclusion in the CCL task, or, in reverse, to convert the MACRO-11 form of the table back to CCL form.

- REMINDER SYSTEM: Three indirect command files -- ENTER.CMD, REMIND.CMD, and WEEK.CMD -- are used to create reminders. ENTER enters a reminder into the list; REMIND will type out the reminders for any day of the week. WEEK lists the reminders for an entire week, starting from today.

- OPLOG: A CCL command or indirect command file (see comments in OPERLOG.CCL) that allow users to quickly make an entry in a system OPERLOG file regarding problems, etc., on the system. Uses EDT V2 and an EDT startup file, OPLOG.EDT.

- CHECK.PAS: Balances a checkbook.

- WHATDOW.C: Types out on TI: what day of the week it is.

- TASKS.CMD: Lists active tasks by terminal number.

[334,1]

This account has various submissions from Wycliffe Bible Translators:

- SPEED gives this capability by plotting on a CRT a graph showing the CPU idle time each second. A graph will be displayed on your terminal.

- SORT.CMD This command file will generate a sort specification file for use with SORT-11. The specification file is generated based on your answers to questions. The terminology used is consistent with the SORT-11 manual. There is extensive help available for any question by pressing the ESC key. Considerable error checking is also done to make sure that any specification file is syntactically correct. An abbreviated dialogue is also available that will make a lot of assumptions for you so you don't have to answer questions dealing with a lot of SORT-11's obscure features.

- VT100 manipulation command files. This is a group of command file that will manipulate the options on your VT100 when you can't remember which bit of which word to change and you can't find your VT100 Programmer's Reference Manual.

* [335,1]

  At LOGIN the VTX task should be run via the LOGIN.CMD file in each UIC. The VTX task will then initialize the attributes, both software and hardware, for the terminal being used. This is done so that a casual user will not be disturbed by any user who might have left the CRT or driver in a non-standard mode. For the user who wants special attributes set at LOGIN, a file VTX.INI may be created that will convey the necessary information.

* [344,1] to [344,67]

  These accounts have Jim Downward's latest KMS Fusion Kit software, all updated for RSX-11M V4.0. Included in the set are the following accounts:

  - [344,1]: Documentation files. All FCSRES features supported in previous releases are now supported by DEC with the PLAS ANSI FCSRES. KMS no longer supports the command and ODL files for the 4K non-ANSI, non-PLAS FCSRES.

  - [344,2]: HELP files specific to using this kit.

  - [344,3]: Command files (.COM) and what not which has been useful to us in migrating from RSX to VMS. Note, that we have command files which do identical things on both our systems. Note, also the super-dooper SPR generator for VMS, and the MEMO and word processing files.

  - [344,5]: Useful command files. MEMO is used for automatic memo generation and filing. WP converts a VT100 to a menue driven word processing terminal. BUILD is an indirect command file which is used in dynamically creating procedure files for PIN. COMPILE and LINK are also supplied as examples of using indirect command files to provide a CCL command. SPR.CMD - The RSX group likes this one. It will produce legible, machine generated SPR addendums. SPR administration will now accept these if you send in a preprinted form with your name attached and 6 or so copies of the addendum. In addition it will keep a file of all SPR's sent which you can use to keep track of where they are in the system.

  - [344,40]: Terminal and disk accounting, privlege allocation, modifications to HELLO, HELP and BYE, account utilities and report programs. Mark Johnson's MAIL Utility for V4.0. Run privlege checking and searching both LIBUIC, NETUIC and SYSUIC on a RUN $ command. SET /LIBUIC for RSX11M (just like M-Plus). Enhancements to ...AT. to implement the /-LI switch.

  - [344,43]: Utilities. An enhanced version of RMC which uses the RPOI$ (Chain) directive if available so RMC can exit at once before command completes and still have exit status passed on. This version of RMC was written to use the new F77 compiler.

- [344,44]: Here is a version of the old RMDEMO which runs on VT100's which are in ANSI mode. This version is almost half as small as the new version and supports reploting and easy exit. In addition, if KMS Accounting is present and active, it provides a dynamic display of the percent of total system utilization. This version (less my mods) was distributed with the field test kit of RSX11M V3.2 and is distributed here courtesy of the the RSX group. It runs only on a RSX11M V4.0 system.

- [344,45]: PDQ -- Program Development Queues. PDQ is a multi-queue command file despooler (using QMG...). Submit a command file for execution. When a program development queue becomes free, the command file executes at your TI: Also included is a Procedure Interpreter (PIN) for use with BATCH. PIN is used as a replacement for ...AT. by PDQ When built with FCSRES, PIN is 4 times smaller than ...AT.

- [344,61]: This version of CCL is a composit of the work of Angle Li at the University of Miami, Paul Sorenson at Michigan State University, and Jim Downward at KMS Fusion. Angle Li edited the code to remove all the funny F4P lables and to comment the code. Paul Sorenson then provided a substanitial rewrite to CCL to provide significantly more flexible command string parsing. V8.0 of CCL will use the RPOI$ directive to chain command lines to tasks. In doing so the OCB's will be relinked and CCL will exit at once since it is not needed to hang around in core to pass back OCB information. This makes it possible to have CCL commands within a command file started by a CCL command. Also CCL was upgraded so that it can be an Alternate CLI for RSX11M V4.0. If it is used as an alternate CLI, it will receive the command line prior to MCR. This means that any MCR command (even for installed tasks) can be trapped and reparsed. Moreover, even with an unmodified MCR, illegal characters can be used as commands. The same CCL task image can be installed both as the catchall task (...CA.) and as an alternate CLI.

- [344,65]: A complete SYSTEM ACCOUNTING package for RSX11M. While oriented towards providing performance measurement, it provides the basis for implementing a versitile and reasonable charge-back accounting system. Some reports are provided which will do this. Please read SYSLOG.DOC.

- [344,66]: This UIC contains programs and command files to aid in measuring system performance. LOAD.CMD will measure system performance for from 1 to 'n' concurrent program development sessions. To fully utilize its features it is best to have implemented KMS accounting, but some rough benchmarks can be done with a vanilla flavor RSX11M V4.0 system. For systems with KMS Accounting, program SNPSHT will produce a snapshot of the system performance statistics which are collected by SYSLOG. SNPSHT produces a command file for setting global parameters (.SETN $PARM val) so that a controling command file can use the values. Also a program (DSKMON) is provided to measure the load on one's system disk

* [350,10]

This distribution kit contains the RSX-11M/M-Plus user monitor files. Please note that this version of the monitor was written for RSX-11M V3.2 and RSX-11M-Plus V1.0; no guarantee is made that it will run under RSX-11M V4.0 or RSX-11M-Plus V2.0.

* [350,11]

OMSI Pascal interface to RMS-11(K) for the PDP-11 Users This account contains all the necessary files, documentation and example programs for interface to RMS-11(K) from OMSI Pascal Version 1 or 2. Since PRM was developed on the (more advanced) RSX system, the installation will be much easier for you RSX users. PRM has never been tested on RSTS/E but using the RSX Run-Time-System, there shouldn't be any problems.

* [350,20]

FFDISK is a formatter for DSD 440 floppies and will format virgin disks. Just run FFDISK (...FFD) and answer the questions. Error codes refer to those defined in the DSD 440 manual.

* [352,2]

This UIC contains Bob Turkelson's files for the TALK program, version V05.11. TALK is a terminal emulation program which enables a user on an RSX-11M V3.2 system with the full duplex terminal driver to communicate with a remote host computer over a single line interface (such as a DLV11 or DL11). The program begins in terminal mode, and has the capability of bi-directional file transfers. See the documentation and program source for details. NOTES.DOC also contains instructions for building TALK.

* [370,1]

This account has submission from the city of Gainesville, Florida - Information Systems Division:

  - TALK: A inter-terminal full duplex (real time) communication tool patterned after the CDC PLATO term-talk feature.

  - SIGN: A sign generator for bulletin boards.

  - PINK: A Pink Panther poster (by J. Noel Carboni). This poster may only be reproduced using a pink ribbon.

  - WOOPS: Our attempt at word processing using EDT and RNO (similar to WPS-11 in that it failed).

* [370,150]

This is an update to the HEX utility that appeared on the Spring 1981 Miami RSX tape under [370,150] and also appeared as 11-480 in the DECUS library. New features include some cosmetic changes, offsetting reads and writes, and the ability to create "even" and "odd" proms for 8-bit proms on a 16-bit data bus. The HEX utility runs under the RSX11M operating system. Its purpose is to manipulate INTEL ASCII Hex-formatted files that are produced by many cross assemblers (or linkers) for microprocessors (8080, Z80, 8085, 68000, etc.).

* [374,1] to [374,2]

At long last, here is a PRELIMINARY binary kit for Decus C which supports floats and doubles. You need FPU hardware, or an emulator. The compiler also emits inline EIS. Many, many bug fixes to the compiler, a complete overhaul of some of it, restructuring of the library, more tools, etc.

Account [374,2] are tools written using the Decus C compiler:

- comm: Compare the contents of two files, indicating what is common to both files, and what is different. comm ? gives help.
- diff: Differential file comparison program, as described in Bell Labs C.S. technical report 41.
- echo: Echo arguments -- used mostly to debug the compiler and run-time system.
- grep: "Global Regular Expression Pattern" -- search files for lines which satisfy an argument pattern. grep ? gives help.
- kwik: Kwik index program. Link with sorts.c.
- l:List on a screen terminal (you hit return every 23 lines).
- mc: A multi-column print utility.
- mp: A full macro processor for C source files, written by R.W. Harper of Rensalear Institute of Technology and originally distributed on a Unix SIG tape. It has been modified to interface with the Decus compiler; it generates the predefined define's and otherwise interfaces just fine now. It supports the full functionality as described in Kernighan and Ritchie.
- nm: Print "namelist" -- prints global symbols for an object module. Lots of options described in the source code, but nowhere else. Does global cross ref's faster then TKB.
- od: Octal dump of a file -- Options described in the source code only. As the RSX library doesn't mangle file attributes, some of the features don't work. od dumps records on RSX, blocks on RT11
- pr: Print with line numbers.

- sortc: Sort a file. Help in sort.rno/sort.doc
- sorts: Sort library routines. Help in the source code. Used by kwik.
- t: File type (on a video screen) utility. Run T and see. Works well on a network virtual terminal. Handles random RMS file formats.
- today: Print today's date in readable English. Uses the following subroutines: datetx (convert date to English), moontx (convert phase of the moon to English), nbrtxt (convert an integer to English), and timetx (convert time of day to English).
- uniq: Print unique lines from a file, uniq ? gives help.
- wc: Word/line/byte counter
- xrf: Cross reference listing for C programs. An excellent tutorial introduction to the C language.

## Working Group News

Elizabeth Bailey
Working Group News Editor
(205) 386-2788

The following is a brief list of working group news from the Atlanta Symposium. More specific details will be published in the coming months. Any time a working group has news for the SIG membership, we would like you to send it to the Multi-Tasker, Working Group Column, so that it will be published.

Several working group chairmen expressed a need for new members and new ideas. A list of working group chairmen was printed in the December/January 1982 issue of the Multi-Tasker. If you would like to participate in any of these groups, contact the appropriate working group chairman.

The RSX UNSUPPORTED VERSIONS group was organized recently for the benefit of those users who, due to small system size or lack of funds, chose not to upgrade and therefore are still running RSX versions 3.0 or 3.1. This group will shortly pick up version 3.2 as well, since indications are that a number of users currently do not plan to upgrade to version 4.0.

This working group has recently come into all the SPR's sent to the Multi-Tasker as a part of the SPR survey. The group will be sorting out the raw material and sending on to the Multi-Tasker the most important information to get published. The Multi-Tasker will handle RSX-11M V4.0 SPR directly. Any future RSX-11M V3.2 SPR's will be forwarded to this group.

The SYSTEM PERFORMANCE AND ACCOUNTING group gathers material on performance measurement and optimization of RSX-11M. It became involved in accounting primarily for the purpose of monitoring system performance. This group has submitted programs on previous SIG tapes and presented sessions at previous DECUS symposia on performance measurement, performance monitoring, and system tuning. An enhancement package for RSX version 4.0 will appear on the Atlanta SIG tape. This group in particular is on the lookout for new members. The chairman of this group, Jim Downward, is moving to VAX/VMS and will not be maintaining the accounting software in the future.

The DECUS LIBRARY group evaluates programs from the DECUS library. Approximately 30 programs have been distributed to working group members and are currently in the process of being evaluated.

The VIRTUAL DISKS group's purpose in life is to maintain and consolidate the implementation of virtual disk packages and keep them compatible. It is currently in the process of working on a virtual disk driver for RSX-11M/M+ which will allow the user to combine multiple devices or files into one virtual volume.

The PROCESS CONTROL group works with computer control of industrial and manufacturing processes. Usually although not always this involves working with the ICS-ICR11 driver. The group provides two functions: a front for discussing the users' needs with DEC, and a user forum for exchanging information. A mailing list is available for the purpose of locating users who have implemented specific applications and specific equipment. Questionnaires were sent to the people in this group in order to update the current status of each group member. The response to this questionnaire was somewhat disappointing.

The questionnaire is reprinted in this issue of the Multi-Tasker so that any interested people who were not on the Process Control group's mailing list will have a chance to respond.

At the Atlanta symposium, several people expressed an interest in reviving the RUNOFF working group. Mike Morgan has agreed to serve as a contact point to see whether enough interest exists to revitalize this group. If you are interested, contact him at this address:

Mike Morgan
City of Virginia Beach
1925 Lanstown Road
Virginia Beach, Va., 23456.

The SIG TAPE COLLECTION working group has several ongoing projects:

1. To get the older tape collections which are in FLX format into BRU format, which most users find easier to work with.

2. To get the earlier symposium tapes into the DECUS library.

3. To build a "best of the SIG tapes" collection.

The possible incompatibility of RSX version 4.0 BRU with version 3.2 BRU has caused some soul-searching for the members of this group. After some debate, version 3.2 BRU was selected as the format for the Atlanta symposium tape. Probably version 4.0 BRU will be used for submissions into the DECUS library and for future symposium tapes.

This working group needs input from the user community in order to make decisions about the material to be included in the "Best of the SIG tapes" collection. They would like to know what programs you think should be included and why. Input from the other working groups, such as the SRD group, is also needed so that the best versions of their software packages will be included.

## Help Yourself

David DiGiacomo
Help Yourself Editor

"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 - Help Yourself, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752.

We would also like to publish the answers to questions. If you can help someone, contact the Multi-Tasker. Your answer will be sent directly to the person in need and published in the next edition.

### Answers to Previous Questions

#### 4010 Graphics Package

Three replies were received for P.A. Johnson's request for a Tektronix 4010 compatible graphics package (March 1981 Multi-Tasker):

Not surprisingly, at least two suitable packages have already been distributed on the SIG tapes. Ken Demers (United Technologies Research Center, Robotics Laboratory, East Hartford, CT 06108, 203 727-7257 or 7240) wrote to suggest the use of his SFGL70 package, which is in UIC [337,30] on the Spring '80 and Fall '81 RSX SIG tapes. Another subroutine package is in UIC [370,30] on the Fall '80 tape.

T. Fong of Canadair sent a product description for Cerritos Computer Services (4320 Atlantic Ave., Suite 1, Long Beach, CA 90807, 213 424-6403) CCSI-PLOT package for the Tektronix 4000 series. CCSI-PLOT is a collection of FORTRAN subroutines which provides basic two-dimensional plotting, axis marking, and software character generation capability. It is based on the traditional CALCOMP primitives.

Manhattan Systems (815 Manhattan Ave., Suite F, Manhattan Beach, CA 90266, 213 372-8544) forwarded information on the GP-10 subroutine package. GP-10 appears to provide the same functions as the Tektronix TCS (Terminal Control System) package -- specifically 2D scaling, rotation, and clipping.

Finally, an extremely sophisticated and comprehensive graphics package is available to non-profit institutions from the National Center for Atmospheric Research (Boulder, Colorado 80303).

## This Month's Questions

### Large Scale Use of FMS-11

Does anyone have experience with using FMS-11 to handle a large number of CRTs? FMS-11 appears to be a very nice product except for one limitation. If we wish to handle 70 CRTs, we must have 70 non-checkpointable FMS tasks installed as well as all our background tasks and DECNET. We currently handle multi-terminal applications by having all screen I/O done by a screen handler. We are impressed with FMS and would like to use it, if possible.

B.J. Checkowy, Supervisor - Systems Development, Saskatchewan Telecommunications, 2121 Saskatchewan Drive, Regina, Saskatchewan S4P 3Y2. Phone (306) 347-2490.

### Source Code Control System

Our Operations staff wishes to control the procedures for updating production software (putting new programs into production). Currently, our development staff does this utilizing CRTs connected to our production systems. Does anyone have, or know of, any librarian procedures or packages for RSX which would provide our Operations staff with the control they feel they require? Has anyone done this at all, and if so, have you had any problems?

B.J. Checkowy, Supervisor - Systems Development, Saskatchewan Telecommunications, 2121 Saskatchewan Drive, Regina, Saskatchewan S4P 3Y2. Phone (306) 347-2490.

### Building Privileged Tasks with FCSRES

We would like to build PR:4 or PR:5 tasks with a 4K non-overlaid FCSRES mapped to APR 7, but the taskbuilder won't let us! This is a real problem for IND/ICP. Does anyone know how to convince TKB not to map to the I/O page?

David DiGiacomo, ECE Dept., Drexel University, 32nd & Chestnut Sts., Philadelphia, PA 19104. Phone (215) 895-2961.

### RSX-11M V4.0 Support for PRESERV

Does anyone have PRESRV built has an RSX utility program? Alternatively, is it possible/easy to have standalone PRESRV recognise a larger subset of Digital disks? Specifically, I need support for the RK07, RM02, RX02, and RL01 as well as some of the disks currently supported.

It was with a degree of incredulity that I learned the PRESRV is being dropped in version 4.0. Surely its ability to treat smaller disks as logical tapes for larger ones is becoming more, rather than less, valuable as winchester disks become more common. Unless of course BRU has been given the capability.

W. David Foubister, Eastman Whipstock (UK) Limited, Denmore House, Denmore Road, Bridge of Don, Aberdeen, AB2 8EF, Scotland. Phone (0224) 703511, ext. 37.

## 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. Send any contributions to Multi-Tasker - Hints and Things, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752.

### Trimming Task Image Files

David DiGiacomo

Drexel University
ECE Department
32nd & Chestnut Sts.
Philadelphia, PA 19104

A little-known task builder feature allows data areas to be eliminated from the task image file. For many programs this leads to significant space savings on the disk. The procedure is quite simple:

1. Determine which data areas do not require load time initialization, and segregate them into one or more PSECTs.

2. Construct an ODL file for the program. Specify an additional co-tree in the .ROOT statement for the data segment.

3. Add a .NAME statement to the ODL specifying the GBL and NODSK option. Note that NODSK informs the taskbuilder not to allocate space in the task image for this segment. But the space is allocated in memory when the task is run.

4. Follow the .NAME statements with .PSECT statements for each of the data areas identified in step 1. The attributes specified for each PSECT must match those assigned in the object code -- check a taskbuilder map if you are uncertain.

5. Add a .FCTR statement specifying the segment name used in the .NAME statement of step 3, followed by the PSECT names from step 4. Label the .FCTR with the co-tree name from step 2.

6. Re-taskbuild the program.

This method is particularly useful for FORTRAN programs which have large COMMON blocks. For example, the following ODL was used to build a slightly modified version of the SORT utility (Spring '81 SIG tape, UIC [300,121]):

```
        .ROOT   SORT/LB:SORTER-SUBS-ERROR-XSUBS,COMMON
;
SUBS:   .FCTR   SORT/LB:ICOMP:IR:LENGTH:WAKEUP
ERROR:  .FCTR   LB:[1,1]SYSLIB/LB:$SHORT
XSUBS:  .FCTR   XINIT-XREAD-XSORT-XWRIT
XINIT:  .FCTR   SORT/LB:INITL
XREAD:  .FCTR   SORT/LB:READIN:WRKOPN
XSORT:  .FCTR   SORT/LB:RCOPY:SORT
XWRIT:  .FCTR   SORT/LB:MERGE:WRITIT
;
        .NAME   COMSEG,GBL,NODSK
        .PSECT  BUFFER,RW,D,GBL,REL,OVR
        .PSECT  COMPAR,RW,D,GBL,REL,OVR
        .PSECT  DIRACC,RW,D,GBL,REL,OVR
        .PSECT  SETUP,RW,D,GBL,REL,OVR
        .PSECT  TABFLG,RW,D,GBL,REL,OVR
COMMON: .FCTR   COMSEG-BUFFER-COMPAR-DIRACC-SETUP-TABFLG
        .END
```

The resulting task is 59. blocks smaller than the original, which is highly significant for those of us running on RK05s.

## 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. Send submissions to Multi-Tasker - SPR, c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752.

### BRU Sorting Bug

From the Editor

The following documents what appears to be a error in the quicksort logic of BRU that has been in since the original release and is also in V4.0 of BRU. This also appears to be the source of incompatability between the V3.2 and V4.0 versions of BRU. The following is from Carl T. Mickelson of Goodyear Aerospace and is an excellent example of how to think through a problem.

What do you do when:

1. You come back from lunch (20 May 1982) to find that the automatic weekly BRU full save, that has run successfully for 6 months, has failed because BRU traps out due to a memory protection failure?

2. A total of 10 to 15 people who regularly use the system are back from lunch too and want to get back to work?

3. You need rapid turn-around on a fix, you cannot afford TSC support (it is too expensive for 1 or 2 calls a year), and a SPR is out of the question since a response is too slow for next week's save?

You find the problem yourself. But how do you accomplish this without the sources for BRU?

Solution:

1. First get on the air as soon as possible by using DSC to compress the disk onto another disk pack, but preserve the original disk to verify a fix for the data dependent error in BRU.

2. Try BRU again and make detailed observations of its performance. The critical observation here was that BRU seemed to work well until about 50% of the disk was written to tape. Then the tape stopped and BRU went into a compute bould loop. Two or three minutes later, BRU crashed due to a memory protection failure with the PC set one byte beyond its allocated memory partition and its stack outside valid stack limits.

3. Review the published patches and try to correlate past problems with the current performance. In article 5.1.17.9 (August 1980), it was noted that a quick-sort algorithm is used to sort file retrieval pointers for faster disk access. Could this problem be related to this sort algorithm, since the CPU seemed compute bound for such a long period? But where to look? Article 5.1.17.9 shows a TKB GBLPAT

designed to fix a stack overflow by changing a constant in module LBNORD.

4. Reconstruct a full BRU map with /MA/CR switches to get full data on all globals defined in the BRU program.

5. Using Tom Getzinger's disassembler available on the DECUS RSX SIG tapes, recreate the psuedo source for the LBNORD module.

6. Studying the code and comparing it with a quick-sort algorithm in Knuth, Volume 3, Sorting and Searching, we observe:

   o Global RPTABL points to a table of retrieval pointers, each 12(10.) bytes long. Global RPEND probably makrs the upper limit of the table. From the BRU map calculate the length of the table to be 12400(5276.) bytes. Thus the table is long enough to hold 1031(537.) entries when the sort algorithm is called. A quick-sort requires a stack whose length is determined by log base 2 of the table length. Each stack entry must hold 2 words, therefore the stack in BRU must be a least twenty words long.

   o Global $QSTAK and $QSTAK + 310(200.) are used as the bases for this dual word entry stack. This stack is 200 words long. A quick-sort algorithm, properly implemented, should NEVER overflow this stack! (What did the GBLPAT of 5.1.17.9 really fix?)

7. We now have some concrete justification to continue to search for a problem. Examining the balance of the pseudo code, we find that the quick-sort subfile sort requests are stacked with:

```
       MOV      X,$QSTAK(R5)
       INC      $QSTAK(R5)
and:
       MOV      X,$QSTAK+200.(R5)
       DEC      $QSTAK+100.(R5)
```

   This second pair of instructions does not stack X-1 as required by the quick-sort algorithm, and the error introduces a data dependent flaw in the sort implementation. It should be noted that the error does not corrupt requests stored in the first half of the stack since the stack should never contain more than 10. words per half and the error decrements data starting 50. words from the base of the stack. It is not suprising then that BRU will sometimes fail to fully save a disk, depending upon the file retrieval pointers on the disk structure.

8. If this is the true cause of the BRU crash, fixing the decrement instruction should prevent a quick-sort stack overflow. Assembling the pseudo source file from the disassembler yields the data necessary to locate the instruction to patch, shown below:

```
       .TITLE  LBNORD
;
; MODIFICATION:
;
```

```
;      01.2 -- FIX SORT LOGIC IN LBNORD (21 MAY 1982)
;              C.T. MICKELSON, GOODYEAR AEROSPACE CORP.
;              AKRON, OHIO 44315 (216) 796-2388
;
       .PSECT
.BLK.=.

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

.=.BLK.
       .END
```

9. Apply the patch as follows:

```
       SET /UIC=[74,60]
       MAC LBNORD.POB=LBNORD.PAT
       LBR LBNORD.OBJ;1=[1,20]BRU.OLB;1/EX:LBNORD
       PAT LBNORD.OBJ;2=LBNORD.OBJ;1/CS:10755,LBNORD.POB/CS:6726
       SET /UIC=[1,20]
       LBR BRU.OLB;0/RP=[74,60]LBNORD.OBJ;2
       TKB @[1,24]BRUBLD
```

10. Try the new version of BRU. In the case documented here, BRU was able to save the same disk on which it crashed earlier. Examination of the original distribution kit shows the bug was in the first release of RSX-11M V3.2 and has not been fixed through autopatch E.

11. The BRU.OLB on the RSX-11M V4.0 distribution kit HAS THE SAME PROBLEM. Hopefully a formal patch will be released by DEC in a timely manner. In the meantime, I think the following patch will fix the problem.

```
       .TITLE  LBNORD
;
; MODIFICATION:
;
;      01.3 -- FIX SORT LOGIC IN LBNORD (21 MAY 1982)
;              C.T. MICKELSON, GOODYEAR AEROSPACE CORP.
;              AKRON, OHIO 44315 (216) 796-2388
;
;              ALSO, REMOVE "FIX" OF 5.1.17.9
;
       .PSECT
.BLK.=.

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

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

```
.=.BLK.
        .END
```

From the Editor

The LBNORD routine checks the data before doing a quicksort to see how many retrieval pointer pairs are out of sequence. It only does the quick-sort if it finds some number larger than a fixed value. In the original RSX-11M V3.2 distribution, this value was 24(20.). The solution in 5.1.17.9 is to patch the fixed value to some higher number, eventually avoiding the quick-sort bug. 5.1.17.9 also stated that changing this value would make the changed version of BRU unable to restore tapes made by the previous version.

In the RSX-11M V4.0 distribution, the fixed value is set to 50(40.). Carl's patch for V4.0 resets the fixed value back to 24(20.), but if you have applied 5.1.17.9, you should probably use that value you selected. In any case, the fix to the quick-sort algorithm should avoid future problems and Digital should have an official fix sometime in the future.

**Various RSX-11M V4.0 Error Log Reporting Problems**

From the Editor

The following list of problems is taken from SPR's posted by Digital in the campground at Atlanta. The actual patch files are quite lengthy (over 25 pages) and are not published here. If one of the problems is critical to your site and the patches have not yet appeared in the Software Dispatch, contact the Multi-Tasker and a copy will be sent to you

1. The command file TUNE.CMD, which is used to 'tune' the Error Log Control File library ERRLOG.ULB, has two problems:

   o The list of supported devices shows the TU77 on the same line as the TE16, TU16, and TU45. This implies that support for the TU77 is automatically included if any of the others are chosen. The TU77 should actually be listed on a separate line, that is, it's support is not tied to the other magtape units.

   o TUNE.CMD does not list support for, or accept, the 11/23 or 11/24 processors.

   The correction file fixes both of these problems.

2. The error log reports for the RA80 disk have the following problems:

   o The COMMAND REFERENCE NUMBER logic is in error.

   o Various lines have misspellings.

   o The message 'Undefined bits set' is occasionally displayed when in fact no bits are set in the displayed field.

   o The MEDIATYPE and shadow condition logic are in error.

   Correction files are supplied to fix these problems in MCSPBD.CNF, MSCPEN.CNF, and MSCPTO.CNF.

3. The RANGE qualifier for the DATE switch accepts two daytime arguments, the first being the beginning day-time of the range and the second being the ending day-time. If the second argument specifies onlya date and no time, such as 12-JAN-82, the time will be defaulted to 00:00:00 rather than rounded up to 23:59:59. The correction file fixes PRS2AM.CNF to fix this problem.

4. Error Log reports of memory parity errors in FULL or REGISTER format fail to print the memory parity CSRs or Cache Control registers. Correction files are supplied to fix this problem in E1134.CNF, E1144.CNF, E1160.CNF, E117X.CNF, and E11XX.CNF.

5. An Error Log report of a Power Recovery in NARROW width and BRIEF format prints as only one line. It should print as two lines so as not upset the page formatting. The correction file fixes DSP6M1.CNF for this problem.

6. The Error Log module for the TS11 incorrectly calculates the alignment of the data being analyzed. This results in an incorrect analysis of the data. The correction file fixes ETS11.CNF for this problem.

7. Various errors happen with disk devices:

   o Error Log reports of peripheral errors in FULL or REGISTER format for MASSBUS devices fail to print all of the registers. The Bus Address Extension 9BAE) and third Control and Status register (CS3) are missing.

- RM80 disk drives are reported as being 'not a Moving Head Device'. Also the high byte of the RMAS register shows 'Undefined bits set' when in fact no bits are set.

- The text for bit 9 in the RPDS register of the RP04/5/6 drives is too long.

Correction files to fix these problems are supplied for EML11.CNF, ERM05.CNF, ERM23.CNF, ERM80.CNF, ERP07.CNF, ERP456.CNF, ERS34.CNF, ET1645.CNF, and ETU77.CNF.

### IND Handling of .DATA Statements

The following SPR for RSX-11M V3.2 IND solves the problem mentioned last month with doing a on-line RSX-11M V4.0 system generation under RSX-11M V3.2. For some command files, Indirect exited with a "AT. -- SYNTAX ERROR" message. This was caused by command files that either had a .DATA statement followed by a carriage return or a tab.

There are two solutions. One is to simply edit the command files mentioned last month and add a space after .DATA statements in question. Or the following SLP command file fixes RSX-11M V3.2 IND to correctly handle the .DATA statements.

```
INDOPN.MAC;2/-BF/AU=[12,10]INDOPN.MAC;1
\
-2,47
        .IDENT  /1.1B/
;
; COPYRIGHT (c) 1981 BY DIGITAL EQUIPMENT CORPORATION.
; ALL RIGHTS RESERVED.
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED
; OR COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.
;
; VERSION:      1.1B
;
;       JRK152  MAKE TAB A VALID.DELIMITER FOLLOWING A .DATA
;       JRK251  ALLOW DATA DIRECTIVE TO BE ONLY TEXT ON LINE
;
%
-217,,/;JRK251/
        TSTB    (R0)                    ; ANY TEXT ON LINE
        BEQ     110$                    ; EQ -> NO, INSERT A CARRIAGE RETURN ONLY
-218,,/;JRK152/
        BEQ     110$                    ; YES, IT MATCHES AS VALID DELIMITER
        CMPB    #11,-1(R0)              ; CHECK IF CHARACTER IS A TAB
-219,,/;JRK152/
        DEC     R0                      ; BACK UP POINTER TO INCLUDE TAB IN BUFFER
-220,220,/;JRK152/
110$:   MOV     R0,R1                   ; YES, COPY BUFFER ADDRESS
-240,240,/;JRK152/
17$:    JMP     INDRD                   ; DONE
-248,248,/;JRK152/
```

```
        BCC     17$                     ; SUCCESS - DONE
```

## Home Directories for RSX-11M/M Plus

Joseph S. Sventek

Computer Science & Mathematics Department
Lawrence Berkeley Laboratory
Berkeley, CA 94720

A common notion in most multi-user operating systems which support permanent file systems is that of a user's home directory. This home directory, which is the default area of a disk (or disks) where the user's file are kept, can be used to keep user-specific information in known files. RSX-11M[+], as well as other DEC operating systems, exploits this fact by permitting the user to create a login.cmd file, with commands for the indirect command file processor to execute whenever the user logs into the system. Other possible uses for the home directory include user-specific initialization files for screen editors, mail systems, teleconferencing systems, etc.

For the home directory concept to be useful, it must be easy for utilities to determine the home directory for the current user. RSX-11M[+] presents a problem in this respect, since no conscious attempt is made to supply this information to the user. The purpose of this note is to provide a means for programs to determine the home directory unambiguously and easily.

For RSX-11M[+], two pieces of information are necessary to construct the home directory: the device specification and the UIC on that device. When logging in, HELLO creates a logical device assignment for the pseudo-device SY:, causing it to point to the disk on which the user's UIC resides. This device assignment is performed with the /LOGIN switch to prevent the user from deassigning it. At any point in time, SY: points to the default device for file specifications. In order to permit the user to find the home directory, it is necessary to create another pseudo-device logical device assignment for each user. The correction file for HELLO which appears in the appendix causes the command:

```
        ASN SY:=HO:/LOGIN
```

to be executed upon the successful assignment of SY: before the terminal is marked as non-privileged. Thus, HO: always points to the home disk.

There are actually three UIC's of importance in the environment of a running task:

1. The default UIC to use if the UIC is omitted from a file specification. This information is kept in the field U.UIC of the TI: UCB and also in the H.DUIC field of each task header.

2. The protection UIC, which is used to determine whether a user can access a file. This information is kept in the H.CUIC field of each task header.

3. The login UIC, which represents the user's login user identification code. This information is kept in the the field U.LUIC of the TI: UCB.

If this was the actual situation, then the determination of the home UIC would be straightforward. Unfortunately, when a privileged user performs a SET /UIC=[g,m] command, the system replaces the contents of U.LUIC in the terminal UCB, in addition to changing the contents of U.UIC. If INSTALL always used the U.LUIC field to initialize the H.CUIC field when installing an image, this would make sense. BUT, INS only uses the U.LUIC field if the user is non-privileged, filling in the H.CUIC field for privileged users from the U.UIC field. Thus one can see a discrepancy in the way the U.LUIC field is used.

Since the U.LUIC field is not used by INSTALL for privileged users, one can apply the patch to SPROVFDT which appears in the appendix. This patch simply eliminates the update of the U.LUIC field for privileged users when performing a SET /UIC command. Thus, the U.LUIC field of the terminal UCB always contains the login UIC. This patch, which was taken from a set of patches by Greg Thompson, does not impact the performance of any RSX utilities.

Although one would expect that this patch would solve the problem, such is not the case. The executive code which performs the GTSK$ directive returns the contents of the H.CUIC header field as the last word of the buffer, not the U.LUIC field of the TI: UCB. Fortunately, since GTSK$ is an optional directive, none of the DEC utilities use it. Therefore, the patch for DRGTK which appears in the appendix causes the appropriate information to be returned to the user.

The author's system has been running successfully with these patches for RSX-11M V.3.2.E since 10/1/81. Utilities exploiting the home directory concept which have been implemented include a mail system, a teleconferencing system and user-dependent editor initialization files.

The appendix contains the correction files for the following source modules:

```
[11,10]DRGTK.MAC
[12,10]HELLO.MAC
[12,10]SPROVFDT.MAC
```

A similar correction to [12,10]SPROV.MAC may be made if the full-duplex terminal driver is not in use. The correction to HELLO has three separate audit trails:

1. JSS000 - causes STOP bit synchronization to be used for the SPWN of ASN if S$$TOP is defined. This supercedes the Digital-supplied correction from auto-patch, RA300.



2. JSS001 - causes @LB:[1,2]SYSLOGIN.CMD to be invoked instead of the user's LOGIN.CMD file. This correction may be left out with no impact on the home directory situation. A sample SYSLOGIN.CMD file is included in the appendix for those wishing to try this feature.

3. JSS002 - the corrections necessary to cause the assignment of the HO: pseudo-device.

Finally, a FORTRAN-callable subroutine which returns the home directory specification to the user as a 0-byte terminated string is included. Note that it will only work if the patches described in this letter are applied.

### DRGTK.COR

```
OU:DRGTK.MAC;2/AU:72./-BF=IN:[11,10]DRGTK.MAC;1
\
-24
;
;       Joseph S. Sventek, 01-Oct-81
;
;               JS001 -- return U.LUIC in protection word if M$$MUP
%
-32,,/; JS001/
        .MCALL  UCBDF$
        UCBDF$                  ; DEFINE UCB OFFSETS
-130,,/; JS001/
        .IF DF  M$$MUP

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

        .IFF
-132,,/; JS001/

        .ENDC

/
```

### SPROVFDT.COR

```
SPROVFDT.MAC;2/AU/-BF=[12,10]SPROVFDT.MAC;1
\
-2,2
        .IDENT  /03.9A/
-89
;       GAT001  DON'T CHANGE U.LUIC ON SET /UIC
;
%
-830,834,/;GAT001/
36$:                            ; REF LABEL
/
```

## HELLO.COR

```
OU:HELLO.MAC;2/-BF/AU:72.=IN:[12,10]HELLO.MAC;1
\
-2,4
        .IDENT  /02.1/
;
; COPYRIGHT (C) 1976, 1978,1979
-24,24
; VERSION:      02.1
-96
;       JSS000  12-DEC-79 DO STOP/WAIT CORRECTLY
;
;       JSS001  10-JAN-80 Hello spawns @lb:[1,2]syslogin.cmd to perform system-
;                         dependent login processing.  The last thing such a
;                         command file should perform is to invoke the user's
;                         login file.
;
;       JSS002  01-OCT-81 perform ASN SY:=HO:/LOGIN
%
-102,102,/;JSS000/
        .MCALL  SPWN$
        .IF DF  S$$TOP
        .MCALL  STSE$
        .IFF
        .MCALL  WTSE$
        .ENDC
-118,,/;JSS002/
HOME:   .ASCII  %ASN SY:=HO:/LOGIN%
HOMLEN=.-HOME
-122,122,/;JSS000/
        .IF DF  S$$TOP
EFNDPB: STSE$   EFN1
        .IFF
EFNDPB: WTSE$   EFN1
        .ENDC
-246,247,/; JSS001/
        .WORD   8.              ;
        .WORD   HELP            ;
-285,290,/; JSS001/
FILNAM: .ASCII  /LOGIN.TXT/     ;
HELP:   .ASCII  /HELP.HLP/      ;
;
;       THE FOLLOWING FOUR LINES MUST BE KEPT IN ORDER
;
LOGINC: .ASCII  /@/             ; START OF INDIRECT COMMAND LINE
DEVNAM: .ASCII  /LB0:/          ; DEVICE FOR [1,2]HELP.TXT AND LOGIN.TXT
DIRNAM: .ASCII  /[1,2]/
LOGINN: .ASCIZ  /SYSLOGIN.CMD/<15> ;
;
;       THE PRECEDING FOUR LINES MUST BE KEPT IN ORDER
;
SYSLOG: .ASCIZ  /@SYSLOGIN.CMD/ ; TELL USER STARTING SYSTEM LOGIN FILE
-415,416,/; JSS001/
```

```
-751,751,/;JSS000/
        DIR$    #EFNDPB         ; WAIT UNTIL ASSIGN COMPLETES
-752,752,/;JSS002/
        MOV     #HOME,SPWDPB+S.PWCA     ; ADDRESS OF ASN SY:=HO:/LOGIN
        MOV     #HOMLEN,SPWDPB+S.PWCL   ; LENGTH OF SAME
        DIR$    #SPWDPB                 ; SPAWN COMMAND
        BCS     170$                    ; IF FAILED, FORGET IT
        DIR$    #EFNDPB                 ; WAIT FOR IT
        BR      170$                    ; CONTINUE ON
-841,841,/; JSS001/
        MOV     #HLPDSP,R2      ; DATA SET POINTER
        MOV     #12.,10(R2)     ; 12 CHARACTERS IN SYSLOGIN.CMD
        MOV     #LOGINN,12(R2)  ; SYSLOGIN.CMD NOT HELP.HLP
-849,849,/; JSS001/
        MOV     #SYSLOG,R0      ; SET COMMAND ADDRESS
/
```

## HOMDIR.MAC

```
        .title  homdir - return home directory spec to user
;+
;       subroutine homdir(buf)
;
;       the home directory spec is returned to the user as a 0-byte
;       terminated string in buf
;
;       this routine depends upon the corrections to DRGTK, SPROVFDT
;       and HELLO published in the multi-tasker (july 1982, vol 16, #1)
;-
ap=%5
buf=2
        .mcall  gtsk$s
        .psect  $r.rod,con,ro,rel,lcl,d
device: .asciz  "HO:"
        .even
        .psect  $r.roi,con,ro,rel,lcl,i
        .enabl  lsb
homdir::
        sub     #32.,sp                 ; space for GTSK buffer
        mov     sp,r0                   ; address of buffer
        gtsk$s  r0                      ; get task parameters
        mov     g.tsdu(r0),r3           ; UIC to format
        add     #32.,sp                 ; restore stack
        mov     buf(ap),r2              ; address of user's buffer
        mov     #device,r1              ; source buffer
10$:
        movb    (r1)+,(r2)+             ; copy character to buffer
        bne     10$                     ; if not null, go again
        tstb    -(r2)                   ; back up one byte
        clr     r4                      ; no leading zeroes, separators
        jsr     pc,.ppasc               ; format UIC into buffer
        clrb    (r2)                    ; null terminate string
        return
        .end
```

### SYSLOGIN.CMD

```
        .ENABLE SUBSTITUTION
        .ENABLE QUIET
.;
.;      put site-specific startup commands in here
.;      for example, if you use the Software Tools mail system,
.;      install POSTMN as ...ITM (for is there mail?) in the tools startup
.;      then place the command
.;      ITM
.;      here, to notify the user whether he has mail or not
.;
        .DISABLE QUIET
        .TESTFILE SY0:'<UIC>'LOGIN.CMD
        .IF <FILERR> NE 1 .GOTO DONE
@SY0:'<UIC>'LOGIN.CMD
.DONE:
```

# Performing RSX-11M V3.2 SYSGENS Under VAX/VMS

Frank J. Nagy

Fermi National Accelerator Laboratory
P.O. Box 500, Mail Stop 306
Batavia, Illinois 60510

Before describing my experiences with a RSX-11M V3.2 SYSGEN done on a VAX-11/780 under VMS, a short description of our system and project will provide useful background information. We are upgrading the control system for Fermilab's high energy physics proton accelerator by replacing the aging Xerox 530's that form the current control system. The new computer system consists of two VAX-11/780's and twelve PDP-11/34's (initially). Unlike the old system in which each of the Xerox 530's was independent and controlled separate accelerator subsystems, the new system will be a network with processes divided among the systems on a functional basis rather than a "geographical" basis.

The system was procured with the understanding that all software development would be done on one of the VAX's (which also stands as a backup for the online control system VAX). On this basis, none of the PDP-11/34's have tape drives or other peripherals needed for a software development system; they only have single or dual RL02 disk systems. Since RSX-11M was the operating system of choice for the 11/34's, we were faced with the need to perform RSX-11M SYSGEN's on the VAX using the RL01/RL02 distribution kit. The remainder of this article was taken from the body of an memo written to instruct other programmers on the project in the intricacies of doing RSX-11M SYSGEN's on the VAX.

This memo discusses the procedures needed to perform an RSX-11M SYSGEN on the VAX. This is based upon actual experience gained by myself over the week prior to July 5, 1981. The distribution kit used was the RL01/RL02 kit as copied from tape onto the four RL02 disks labeled RSXM26, MAPSRC, UNMSRC, and RLUTIL. Since we are only interested in a mapped system, the UNMSRC disk is not used in the SYSGEN. In point of fact, only the RSXM26 and MAPSRC disks are really used. Since our VAX is equipped with dual RL02 disk drives, the SYSGEN was done using RL02 disks only. In the future (distant?), we may be able to SYSGEN an RSX system onto an RL02 from an RM03 disk pack but for now we shall have to use both RL02 disk drives while doing a SYSGEN.

The first step is to prepare a copy of the RSXM26 disk onto another RL02 disk. This additional disk will be the one on which the new system is generated. The RSXM26 disk should not be used for SYSGEN's except as the input for a copy operation. The disk can be copied on the VAX using DSC1. This is done by loading the disks into the drives and then mounting them with VMS MOUNT commands. The RSXM26 input disk should be loaded with WRITE PROTECT, and can be additionally mounted with the /NOWRITE qualifier. The output disk must also be mounted, but as a foreign volume using the /FOREIGN qualifier. The commands to mount the volumes and run DSC1 are thus (assuming DCL is the current Command Line Interpreter):

```
$ MOUNT/NOWRITE dln:  RSXM26
$ MOUNT/FOREIGN dlm:
$ MCR DSC1 dlm:=dln:
```

Before the SYSGEN can begin, a VMS system parameter must be modified. This parameter is CLISYMTBL and governs the size (in pages) of each process' Command Line Interpreter symbol table. The normal setting for CLISYMTBL is 20; a value of 36 is adequate for doing a SYSGEN. The parameter is modified by logging in as a privileged user and running the VMS SYSGEN program. This operation requires the CMKRNL privilege. The commands used to modify CLISYMTBL are thus:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>USE ACTIVE
SYSGEN>SET CLISYMTBL 36
SYSGEN>WRITE ACTIVE
SYSGEN>EXIT
```

I then logged out since to do the RSX SYSGEN, you must have MCR as your Command Line Interpreter. Also, I believe that the CLISYMTBL parameter only takes effect when a process is created (i.e., login time).

To start the RSX SYSGEN, I logged in with the "/CLI=MCR" qualifier after my username. This established MCR as the CLI for this process. At this time our system-wide login command file (SYLOGIN.CMD) was invoked to give a greeting, to set the default file protection, and to assign the logical names TEC and

MAK to invoke the TECO editor. My own LOGIN.CMD file was then started. Listings of my LOGIN.CMD and the RSXSYSGEN.CMD files are included with this memo. The RSXSYSGEN file was executed when I responded Y (YES) to the "Is an RSX-11M SYSGEN being done?" question. The RSXSYSGEN.CMD file was basically copied from the example in the *VAX-11/RSX-11M User's Guide* with the following changes:

1. Logical names LB0 and SY0 are assigned to LB and SY rather than directly to the target disk.

2. Logical name TKB is assigned to SYS$SYSTEM:BIGTKB.TSK. This is the RSX task builder (not the VMS version TKB.EXE in the same area) copied to our RM03 to allow faster overlays. As with DEC's example, this version will also allow task images to have a default file type of .TSK (rather than .EXE for the VMS TKB).

3. The logical names MAC, EDT, LBR, and PIP are assigned such that invoking one of these utilities will get the version from the VMS system disk. Like the change for TKB (above), this also allows the overlays to be loaded into memory faster. However, using the VMS MAC.EXE allows the assembly listings to be placed on magtape or another disk (DRC2: in our case).

4. Finally the SYSPRV, LOG_IO, and CMKRNL privileges are set.

Note that only a privileged user with the SETPRV privilege can use this procedure to do an RSX SYSGEN. Also note that the task builder being used is the RSX BIGTKB which will *not* handle references to magtape or VMS device names (i.e. DRC2:). For this reason, I had the task builder maps placed on the target disk in [1,34]. Since the target system disk is an RL02, there is plenty of space.

After the login procedures complete, you must mount the copied RSXM26 disk. If you want the task builder maps spooled and they are to be written to [1,34] on this disk, then the disk must be mounted shareable so that the print symbiont can access the files. The commands to mount the disk and start the RSX SYSGEN are:

```
≥ MOUNT{/SHARE} dln:RSXM26
≥ SET /UIC=[200,200]
≥ @SYSGEN
```

You are now off and running in SYSGEN Phase I. Phase I should proceed exactly the same as doing a SYSGEN online under RSX-11M. Remember to answer NO to the question about running under the standalone RSX system. The Executive Sources are on the MAPSRC disk which I mounted on the other RL02 disk drive. If the MAPSRC disk itself is used, rather than a copy, several files must be cleaned-up after the SYSGEN (more about this later).

Phase I should proceed normally, just as if done online under RSX. Before starting Phase II, the MAPSRC disk should be dismounted using the DMO command (but not physically dismounted). This will prevent an error message when MAPSRC is again mounted during Phase II. I had to modify SYSGEN2.CMD in order to have it perform properly. Without my patches, all the privileged task builds would fail because TKB could not find the command files. This was due to the method used to swap the assignments of LB and SY just before this point. In the current SYSGEN2.CMD file (Autopatch Level E), this will not work on the VAX and both SY and LB are assigned to the target disk rather than having SY assigned to the MAPSRC disk. I have modified the SYSGEN2.CMD file on the RSXM26 disk using a modified SYSGEN2.COR corrections file on our AUTOPATCHE1 disk so that a dual RL02 SYSGEN can be done on the VAX. The versions of SYSGEN2.CMD on the MAPSRC and UNMSRC disks have not been updated in this way! A listing of the SLP input file to update SYSGEN2.COR can be found at the end of this article.

During Phase II of SYSGEN I attempted to edit the RSXBLD.CMD and SYSVMR.CMD files. I had the EDT editor abort with an access violation which seems to have been caused by having the UIC set to the wrong value. Or at least that appeared to be the cause in one case. In another case, I just tried again to edit (with EDT) RSXBLD.CMD and everything worked properly. I'm not really sure what the problem is here or if there is any but you should be aware of possible difficulties. After Phase II of SYSGEN was over and before continuing with anything else, I built two additional privileged tasks. The first of these was BIGIND.TSK since the version on the disk is built for the standalone distribution system. If you plan to do any SYSGEN's using the resulting RSX system, BIGIND must be re-built. Otherwise BIGIND.TSK should be deleted from the disk. The other task build was for MC2, the MCR catch-all task which implements the additional DCL-like commands. The following commands can be used to build these two tasks (after Phase II, MAPSRC is still mounted):

```
≥ SET /UIC=[1,24]
≥ ASN dlm:=IN:    !assign to MAPSRC disk
≥ @MC2            !assemble MC2
≥ ASN dlm:=SY:    !assign to MAPSRC disk
≥ ASN dln:=LB:    !target disk (already assigned)
≥ ASN dln:=TK:    !target disk for task image
≥ TKB @BIGINDBLD  !build BIGIND
≥ TKB @MC2BLD     !build MC2
```

The MC2 listing file and the task builder maps are output to MP:[1,34] and MP is left assigned to the device for the privileged task build maps.

After these were done, I copied MCR.OLB from [1,24] on the MAPSRC disk to [1,24] on the target system disk. This will be needed for a couple of task builds done for additional drivers and utility software from the KITSMISC (software kits) disk and is useful to have on a development system disk. Note that besides the addition of MC2 to MCR.OLB, the modules SDSOV, SETOV, and SPROV were replaced by different versions as the system I generated uses the full-duplex terminal driver (see below about restoring the MAPSRC disk).

Phase III of the RSX SYSGEN can be skipped as the non-privileged task images can be copied from the KITSMISC Software Kits disc and the User Mode Diagnostics are useless with our systems. Instead, if desired, the MAPSRC disk can be dismounted and the RLUTIL disk mounted and the contents of [2,300] be copied from the RLUTIL disk to [2,300] on the target system disk. The UFD [2,300] contains the Installation Verification Program as described in the RSX SYSGEN manual.

At this point the SYSGEN was complete as far as the VAX was concerned. Before removing the target system disk from the VAX, I used the KITSMISC disk to add some new SYSLIB routines and some new libraries to [1,1] (including the F4P libraries and object code files). I also copied the non-privileged utility tasks from KITSMISC and generated some additional privileged utility tasks and device drivers (including the PCL drivers).

The remainder of the SYSGEN activities now move to a PDP-11. I have fixed our RSXM26 disks so that when booted with the hardware bootstrap, the mapped standalone system is started from [1,54]RSX11M.SYS;1. This is needed since Autopatch removed [1,50] as a mapped system was to be generated. The standalone system has also been saved with the BOOT task installed so that the proper version of BOO.TSK is used with the standalone system. After booting the standalone system, I typed control-Z when asked for the date and time and then typed "BOOT [1,54]RSX11M" to boot the system I just created. This system was started in XDT (selected as a SYSGEN option) and a "G" started RSX itself. The system was started with "SAV", but the boot block was not overwritten yet. After this system was rebooted, following the SAVe, I executed the SYSINS.CMD file to install the non-privileged utilities (such as BIGMAC and BIGTKB, etc.). After this executed properly, I saved the system with "SAV /WB", rewriting the boot block so that a hardware bootstrap operation would boot the new system. At this point then, the [1,54] UFD was purged of all old versions (including the standalone RSX11M.SYS file) and all unneeded device driver files. The final task was then to run the Installation Verification Procedure to check out the system before replacing the old RSX Development System with the new version.

Since I used the MAPSRC disk without first making a copy of it, I had to remove some files added by the SYSGEN. In [1,54] the new version of RSX11M.STB was deleted. Note that the NEW version was deleted, a purge operation was not done! In [1,24] the SYSPAR.CMD file was deleted and MCR.OLB was replaced. The easiest way to fix MCR.OLB is to create a new version before doing the SYSGEN and then delete the new version afterwards. When the full-duplex terminal driver is created, several object modules in MCR.OLB are replaced from the object file MCRTTFD.OBJ (also in [1,24]). After the SYSGEN, the old versions may be restored from MCRTTHD.OBJ so that subsequent SYSGEN's for the half-duplex terminal driver will be correct. The recommended procedure is to create a new MCR.OLB and then delete it after the SYSGEN. This keeps MCR.OLB from growing due to the many LBR replace operations.

The disks used for this RSX SYSGEN contained an patched version of the RSX distribution kit. For this SYSGEN, Autopatch Level D was run on the VAX to bring the "virgin" distribution kit up to the latest patch level. For the most part, Autopatch Level D worked correctly after running a preparation file on the VAX. This preparation file (VAXPREP.CMD) is listed below. Its major purpose is to created the needed UFD's on the RSXM26 disk because PIP (on the VAX) cannot create a UFD. Subsequent to this SYSGEN, Autopatch Level E came out but was not nearly so easily installed using the VAX. Two major problems in the AUTOPATCH.CMD file were fixed in order to allow Autopatch E to be done using the VAX:

1. The use of <SYDISK> on the VAX was replaced to avoid references to "SYn:" when "ddn:" (target disk) was expected. Note that this is the same problem that was corrected with the SYSGEN2.CMD file.

2. All the LBR lines which extracted files from universal libraries of the form:

   ```
   LBR OU:file.COR=[uic]CONCAT.ULB/EX:file
   ```

   were fixed since some LBR bug caused OU to be flagged as an illegal device. The fix involved replacing OU by a symbol replacement operation where the symbol was defined to be the name of the target disk. This name had to be the physical device name (i.e. "DL1:") rather than a logical name in order for LBR to accept it.

With these changes then, Autopatch E was applied to the RSX kit which was then used by several other people to perform RSX SYSGEN's with no major problems.

Since my initial attempts at doing the RSX-11M V3.2 SYSGEN on the VAX, several others have used my procedures to perform additional SYSGEN's on the VAX. At this time we have received the RSX-11M V4.0 distribution but have not yet attempted to SYSGEN an RSX V4 system using the VAX. We have not even tried to copy the RSX V4 tapes onto RL02 disks as I expect we will have problems using DSC1.EXE to do so. Hopefully we will attempt to generate an RSX V4 system this summer but based on reports of the difficulty of doing a V4.0 SYSGEN on a V3.2 RSX system I expect we will have many problems trying to do a V4.0 SYSGEN on the VAX.

## Listing of file RSXSYSGEN.CMD

```
.;
.; RSXSYSGEN.CMD
.;
.; Used to make the logical name assignments to do an RSX SYSGEN
.; on the VAX under VMS.  From the VAX-11/RSX-11M USER'S GUIDE
.;
.; Modified:
.; FJN   01-Jul-81       Make sure MAC used from SYS$SYSTEM:
.; FJN   03-Jul-81       Straighten out assignments and
.;                       make sure utilities used from SYS$SYSTEM
.; FJN   05-Jul-81       Use BIGTKB.TSK on SYS$SYSTEM:
.; ADT/KJC 25-Jan-82     Modify message concerning spooled maps
.;                       and the MOU/SHARE command for clarity.
.;
.ENABLE SUBSTITUTION
.;
.; Get RSX target device is not entered as a parameter
.;
.IF P1 EQ "" .ASKS P1 Target device name
.;
.; Make sure the device name is ended by a colon
.;
.SETS TEMP P1[<STRLEN>:<STRLEN>]
.IF TEMP NE ":" .SETS P1 P1+":"
SHOW SYMBOL P1
.;
.; Set default to target device
.;
SET DEFAULT 'P1'
.;
.; Assign SY, SY0, LB, and LB0 to target disk
.;
ASN 'P1'=SY:
ASN SY:=SY0:
ASN 'P1'=LB:
ASN LB:=LB0:
.;
.; Equates MP0 and MP, also TK0 and TK
.;
ASN MP:=MP0:
ASN TK:=TK0:
.;
.; Assigns equivalence names for TKB and VMR
.; BIGTKB.TSK was directly copied from an RSX system.
.;
ASN SYS$SYSTEM:BIGTKB.TSK=TKB
ASN 'P1'[1,54]VMR.TSK=VMR
.;
.; Make sure utilities used from SYS$SYSTEM:
.;
ASN SYS$SYSTEM:MAC=MAC
ASN SYS$SYSTEM:EDT=EDT
ASN SYS$SYSTEM:LBR=LBR
ASN SYS$SYSTEM:PIP=PIP
.;
```

```
.; Set process privileges needed to do a SYSGEN
.;
SET PROCESS/PRIVILEGES=(SYSPRV,LOG_IO,CMKRNL)
;
; Note that the baseline system TKB is used.   The maps
; cannot be sent to a non-RSX device or to magtape.  They may
; be put on the target disk ('P1').  It must be mounted
; SHARE'd to allow the maps to be spooled to the printer:
;
;        > MOU/SHARE 'P1'RSXM26
;        > SET /UIC=[200,200]
;        > @SYSGEN
;
```

## Listing of file [NAGY]LOGIN.CMD

```
.;
.; [NAGY]LOGIN.CMD
.;
.;       Login command file for use with the MCR CLI
.;
.;       Always set SYSPRV privilege
SET PROCESS/PRIVILEGE=SYSPRV
.;
.; Ask if an RSX SYSGEN is to be done.
.;
.ASK SYSGEN Is an RSX-11M SYSGEN being done
.IFT SYSGEN @SYS$SYSDISK:[SYSMGR.UTILITIES]RSXSYSGEN
.;
.; Assign some additional tasks
.;
ASN SYS$SYSTEM:PIP=CPY
.;
```

## Listing of VAXPREP.CMD

```
.;
.; [200,200]VAXPREP.CMD on Autopatch disk.
.;
.; Modifications:
.;
.; 14-Oct-81     FJN&KJC Brought up to level E by adding UFD [64,60] and
.;                       INIT for scratch disk.
.; 16-Oct-81     FJN     Replace SET PROT by PIP to change UFD protections
.;
.ENABLE SUBSTITUTION
.ENABLE GLOBAL
;
;
; AUTOPATCH Rev. E Preparation file for VAX/VMS.
;
```

```
; This command file is run on the VAX (under the MCR CLI) to prepare
; for performing an RSX Autopatch on the VAX.  When logging in, add
; the qualifier /CLI=MCR after your name to use MCR as the command
; line interpreter.  You must run on an account with SYSPRV (or
; SETPRV to enable SYSPRV) and CMKRNL to allow the user of SET UIC.
; Mount the Autopatch disk and then execute this command file
; via:
;        @ddu:[200,200]VAXPREP
;
;
.ASKS APDEV What device (ddu:) is the Autopatch disk mounted on?
.;
1$:      .ASK YESNO Do you want to initialize an ODS-1 scratch disk?
.IFF YESNO .GOTO 2$
.ASKS [1:9.] SCRLBL Enter the volume label for the scratch disk
.ASKS TMPDEV When ready, enter device (ddu:) with scratch disk loaded
INIT/STRUCTURE=1/HEADERS=800. 'TMPDEV' 'SCRLBL'
MOU 'TMPDEV' 'SCRLBL'
.ASK YESNO Did the scratch disk mount successfully?
.IFF YESNO .GOTO 1$
DMO 'TMPDEV'
.;
2$: .ASKS SYSDEV When ready, enter device (ddu:) with RSXM26 disk
MOU 'SYSDEV'RSXM26
.ASK DIDIT Did the RSXM26 disk mount successfully?
.IFF DIDIT .GOTO 2$
;
; Setting logical names SY: and SY0: (RSX default disk) to 'SYSDEV'.
;
ASN 'SYSDEV'=SY:
ASN 'SYSDEV'=SY0:
;
; Create UFDs for Autopatch files.
; Allow them to be "delete-able" so they can be removed by
; the (modified) Autopatch.
;
UFD 'SYSDEV'[01,60]/UIC=[01,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[11,60]/UIC=[11,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[12,60]/UIC=[12,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[13,60]/UIC=[13,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[14,60]/UIC=[14,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[15,60]/UIC=[15,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[22,60]/UIC=[22,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[31,60]/UIC=[31,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[32,60]/UIC=[32,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[33,60]/UIC=[33,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[36,60]/UIC=[36,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[41,60]/UIC=[41,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[42,60]/UIC=[42,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[45,60]/UIC=[45,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[47,60]/UIC=[47,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[50,60]/UIC=[50,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[52,60]/UIC=[52,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[55,60]/UIC=[55,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[62,60]/UIC=[62,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[63,60]/UIC=[63,60]/PRO=[RWED,RWED,RWED,RWED]
```

```
UFD 'SYSDEV'[64,60]/UIC=[64,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[70,60]/UIC=[70,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[74,60]/UIC=[74,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[106,60]/UIC=[106,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[112,60]/UIC=[112,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[121,60]/UIC=[121,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[163,60]/UIC=[163,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[200,60]/UIC=[200,60]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[01,64]/UIC=[01,64]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[12,64]/UIC=[12,64]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[14,64]/UIC=[14,64]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[15,64]/UIC=[15,64]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[25,64]/UIC=[25,64]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[50,64]/UIC=[50,64]/PRO=[RWED,RWED,RWED,RWED]
UFD 'SYSDEV'[121,64]/UIC=[121,64]/PRO=[RWED,RWED,RWED,RWED]
PIP 'SYSDEV'[0,0]*.DIR/PR/SY:RWED/OW:RWED
.;
;
; Setting default directory and device to 'SYSDEV'[200,200].
;
SET /UIC=[200,200]
SET DEFAULT 'SYSDEV'[200,200]
.;
PIP SY:/NV='APDEV'AUTOPATCH.CMD
;
; Dismount the Autopatch disk (DMO 'APDEV') and then
; do "@AUTOPATCH" to begin the Autopatching process.
;
```

SLP correction file for [200,60]SYSGEN2.COR

```
-12
.; FJN001 -- Correct ASN's and use of <SYDISK> for VAX-based SYSGEN's
.;
.; FJN002 -- Permit editor selection rather than use EDI only
.;
.; FJN003 -- Add SET DEFAULT commands for doing SYSGEN on VAX
.;
<-115,115,/;FJN001/
.SETS TEMPS <SYDISK>+"'<SYUNIT>':"
.IF <SYSTEM> EQ 5 .SETS TEMPS "'F$LOGICAL(<SYDISK>)'"
.IFT $RL .IF $PRV NE "'TEMPS'" .SETT $2RL
<-128,,/;FJN001/
.IF <SYSTEM> EQ 5 .GOTO ONVAX1
<-131,131,/;FJN001/
.GOTO ENVAX1
.ONVAX1: .SETS TEMPS "'$PRV'"
.IF $PRV EQ "SY:" .SETS TEMPS "'F$LOGICAL($PRV)'"
.IFF $MLH ASN 'TEMPS'=LB:
.SETS TEMPS "'F$LOGICAL(<SYDISK>)'"
.IFF $MLH ASN 'TEMPS'=TK:
.ENVAX1: .IFT $2RK .OR .IFT $2RL .SETT $2DSK
<-137,138,/;FJN002/
.ASKEDI: .ASKS FJNEDI 5A.  Which editor is preferred (EDI,EDT,TEC)
.IF FJNEDI EQ "" .SETS FJNEDI "EDT"
```

```
.IF FJNEDI NE "EDI" .IF FJNEDI NE "EDT" .IF FJNEDI NE "TEC" .GOTO ASKEDI
.SETT EDITX
.IFNINS 'FJNEDI' .SETF EDITX
-17
<-214,216,/;FJN002/
.IFF $MLH .IFNINS 'FJNEDI' INS SY:$'FJNEDI'
.IFF $MLH 'FJNEDI' ISBLD.CMD
.IFF $MLH .IFF EDITX REM 'FJNEDI'
-48
<-704,705,/;FJN002/
.IFF $MLH .IFNINS 'FJNEDI' INS SY:$'FJNEDI'
<-707,707,/;FJN002/
.IFF $MLH .IFF EDITX REM 'FJNEDI'
<-754,754,/;FJN001/
.700:  .IF <SYSTEM> EQ 5 .GOTO ONVAX7
.SETS TEMPS <SYDISK>+"'<SYUNIT>'"+":"
<-757,757,/;FJN001/
.GOTO ENVAX7
.ONVAX7: .SETS TEMPS "'F$LOGICAL(<SYDISK>)'"
.SETS LBDISK "LB"
.SETS TEMPS2 "'F$LOGICAL(LBDISK)'"
.IF TEMPS2 EQ "" .SETS TEMPS2 $PRV
.ENVAX7: .IFF $MLH .IFT $2DSK ASN =LB:
<-760,,/;FJN003/
.IF <SYSTEM> EQ 5 SET DEFAULT SY:'<UIC>'
<-831,831,/;FJN001/
.KENNER: .IF <SYSTEM> EQ 5 .GOTO ONVAXK
.SETS TEMPS <SYDISK>+"'<SYUNIT>'"+":"
<-834,834,/;FJN001/
.GOTO ENVAXK
.ONVAXK: .SETS TEMPS "'F$LOGICAL(<SYDISK>)'"

.SETS TEMPS2 "'F$LOGICAL(LBDISK)'"
.ENVAXK: .IFF $MLH .IFT $2DSK ASN =LB:
<-837,,/;FJN003/
.IF <SYSTEM> EQ 5 SET DEFAULT SY:'<UIC>'
-56
<-1017,1019,/;FJN002/
.IFF $MLH .IFNINS 'FJNEDI' INS SY:$'FJNEDI'
.IFF $MLH 'FJNEDI' SYSVMR.CMD
.IFF $MLH .IFF EDITX REM 'FJNEDI'
<-1025,1025,/;FJN001/
.IFF $MLH .IF <SYSTEM> NE 5 ASN SY:=LB:
.IFF $MLH .IF <SYSTEM> EQ 5 ASN 'F$LOGICAL(<SYDISK>)'=LB:
<-1050,1050,/;FJN001/
.IF <SYSTEM> NE 5 .SETS DV "'<SYDISK>''<SYUNIT>'"
.IF <SYSTEM> EQ 5 .SETS DV "'F$LOGICAL(<SYDISK>)'"
<-1111,1111,/;FJN002/
.IFF $MLH .IFF EDITX .IFINS 'FJNEDI' REM 'FJNEDI'
/
```

# RSX-11M to VAX/VMS Transmigration

James G. Downward
Pamela C. Vavra

KMS Fusion, Inc.
P.O. Box 1567
Ann Arbor, Michigan 48104

### VMS/RSX Executive Library

At present, we are interested in creating a subroutine library of VAX calls which mimic the RSX executive calls. Our goal is to create a subroutine library which will enable a user to compile and link a Fortran task using native mode VMS utilities and without having to make any coding changes from the RSX code. If other users are interested in this project, we would be interested in hearing from them. Perhaps by trading subroutines the time to complete the project could be significantly shortened. You can contact us at the address above or call (313) 769-8500.

### ABSTRACT

KMS Fusion recently installed a VAX system to act as a host for program development for its real time systems, to handle non-time critical processing tasks, and to support image processing. Within a short time frame (four months) major applications had been migrated to the VAX, and the VAX was being used to develop application tasks for the real time PDP-11 systems running RSX-11M. This paper discusses the results of the first four months of transmigration work comparing our experiences and difficulties with the initial assumptions upon which the project was started.

## INTRODUCTION

KMS Fusion has four PDP-11 mini-computers engaged in data acquisition, process control, and program development activites. The first PDP was installed in 1976. Years of steadily increasing load on the systems had been accompanied by major efforts at enhancing their performance. Finally, it became obvious that short of buying additional computing power, little more could be done to handle the increasing demands on the systems. To meet the immediate processing requirements and to allow for future expansion, we had the choice of either procuring several 16-22 bit processors capable of a mixture of program development and real-time control, or procuring a single, large, program development machine, and dedicating the smaller PDP-11's to real-time tasks.

The number one requirement we faced was that the selected system must provide program development capability for PDP-11 based real-time data acquisition and process control functions. This meant using either PDP-11 or VAX processors. The second requirement was that the system must support two nearly independent Image Processing stations. The I/O intensive demands of image processing tasks required MASSBUS disks and at least 22-bit addressing capability. Both the PDP-11/70 and the VAX-11/750 met these requirements and were within budget constraints.

The principal advantage of a PDP-11/70 configuration was little-to-no code conversion, we could use existing device drivers for foreign peripherals. Moreover, users would stay with a "comfortable" and "familiar" operating system. The principal disadvantage was the antiquity of the PDP-11/70 architecture; it's the same vintage as our existing 11/45. Since it was getting harder for DEC field service to keep our PDP-11/45 working reliably we might experience similar difficulties with a PDP-11/70. Moreover, the time involved in "operating system maintenance" for our RSX11M systems was already rather high. The prospect of two more systems to worry about SYSGENing and AUTOPATCHing was not welcomed.

The advantages of procuring a VAX included cost (a single VAX-11/750 costs less than two PDP-11/70 systems), strong DEC support for VMS, an operating system which was purported to be simpler than RSX to maintain, and superior native-mode software development tools. Moreover, the salesman "guaranteed" we could use the AME to develop and debug programs for our RSX systems. On the other hand, procuring a VAX meant we'd have to install a network in order to talk to the acquisition computers. We had no experience with DECnet, and the rumors heard at DECUS were somewhat discouraging. Even assuming that the network was easily set up, using the VAX would require two independent departments with divergent priorities to share its resources. This would require an improved level of communication and coordination between the departments.

Clearly, the choice between the two options was not easy. Eventually, we decided to procure a VAX-11/750 with two RM03 disk drives, three DZ-11 terminal interfaces, a TS11 tape drive, and four DMR11 interfaces for networking to three of the four existing PDP-11 minis. The order was placed at the end of September, with delivery of our system scheduled for December.

## PRE-DELIVERY PREPARATION

Because the time from anticipated delivery to the time at which the system must be fully functional was rather short, a significant amount of front-end planning had to be done. An inter-departmental VAX working group was formed to oversee the preparations, training, and management issues relating to procuring a single machine to be used by several groups. From the beginning we attempted to clearly define the roles to be played by the PDP-11's and the VAX, and the work which would be done on each. Specific guidelines included:

1. Place all non-time critical functions on the VAX:
   a) Editing.
   b) Document preparation.
   c) Program development for RSX11M systems (where possible).
   d) Data entry, analysis and reporting for the real-time systems.
   e) Image acquisition, processing and analysis.

2. Data analysis and reporting tasks used by the real time acquisition systems must be able to run on and work from any node on the network. If the network goes down, the individual node must be TOTALLY able to function alone.

Surprisingly, one of the issues most persistently raised by management during the months prior to delivery was that of "How to manage the VAX". While we had been running our PDP11's for years without really "managing" them, the fact that the VAX was viewed as a large, almost main-frame class processor, made many people worry that the VAX would require a significant portion of time and manpower to just "manage it" as a service.

The first significant step in our pre-delivery preparations was to procure ALL (or nearly all) available technical literature on the VAX and how to use it. This literature included:

1. The VMS manual set. Manual sets for all purchased layered products (DECnet,VAX FORTRAN, VAX BASIC, DATATRIEVE)

2. VAX Architecture Handbook

3. VAX/VMS Internals and Data Structures manual.

4. Self paced instruction manuals including:
   a) VAX Concepts course
   b) VAX Instruction Set course
   c) VAX/VMS Utilities and Commands course
   d) VAX FORTRAN/MACRO Programer course
   e) VAX BASIC Programer course
   f) Program tapes with examples for every course that had them.

5. All past VAX/VMS SIG symposium tapes

The amount of literature was overwhelming! We started with the VMS manual set, worked our way through it several times, and then read the rest of the manuals and course materials. This process was eased considerably by reading the manuals with specific goals in mind:

1. Remember where major topics are discussed, where problems are pointed out and where examples are given. Don't worry about understanding everything.

2. What are the applications we want to migrate? What should be the priority for migrating applications?

3. How does VMS affect, help, hinder the applications we want to migrate?

4. What are going to be problem areas with
   a) Managing VMS?
   b) Training users?
   c) Running programs under the AME?
   d) Converting programs to native mode?

5. What are the easiest, lowest risk things to do first?

To get a feeling for how VMS really runs, we attended some of the VMS sessions at the Fall 1981 DECUS meeting and talked to users about their experiences. We also obtained back copies of the Page Swapper so that we could read about real issues, problems, etc. that VMS has, and visited three local VAX installations. The discussions we had with users at the local VAX installations confirmed our assumptions that the VAX/VMS would be both easy to get up and running and to maintain.

## INITIAL SYSTEM INSTALLATION AND CHECKOUT PLANS

By the time December rolled around, we had a pretty good idea of of the timetable we would try to follow. Because work on the PDP-11 systems had to continue unabated, the migration process would have to be gradual. As time allowed, we would gently ease users and their applications off of the PDP-11s and on to the VAX. Specifically, our implementation plan was:

1. Attempt the simple things first and be sure operational procedures were solid prior to any user application becoming committed to using the VAX.

   1. Generate the operating system and install software. Coming from the RSX world we grossly overestimated the difficulty of this process on VMS.

   2. Create commonly used command procedures to help the system manage itself and to ease user migration from the PDP-11's. Such procedures include:
      a) System startup procedures
      b) System default logon procedures
      c) SPR generator
      d) Backup command procedure
      e) Word processing and memo filing procedures
      f) Operator command procedures to allow operators to
         a) Add accounts to the system
         b) Incrementally backup the system and keep a file of backup dates.
         c) Shut down the system.
      g) Migrate simple data base entry tasks for the engineering drawing file data base and the security access data base.
      h) Migrate document preparation.
      i) Examine the VMS software available on the SIG tapes.
      j) Create site specific help files.
      k) Investigate accounting and resource monitoring.

2. Following a month of familiarization with the system, we would bring up DECNET on the VAX and RSX systems. Once DECnet was running, we would migrate our acquisition system report programs over to the VAX and access our data files across the network. This would give us experience in using the network, in using the VAX to develop and maintain a collection of programs for the PDP-11's.

3. Migrate the image processor over to the VAX. This effort would involve rewriting 300 blocks of assembly language routines in Macro-32 and converting 3-4000 blocks of FORTRAN source code to running on the VAX.

## VAX/VMS AS SEEN THROUGH RSX COLORED GLASSES

### System Generation

The system was delivered right before Christmas and was installed the next week. We were amazed at how simple it was to get the operating system up and running. Generating the system and installing all the layered products took less than a day to accomplish. The single most significant problem we encountered was getting the print spooler to output to LA120's. Since the initial installation, system maintenance has taken almost no time despite the fact that we have updated VMS V2.4 to VMS V2.5, installed DECnet, and reinstalled updated releases of FORTRAN and Datatrieve.

### System Reliability, Bugs, and SPR Responses

VMS seems to present a stable, reliable, and responsive environment for the user. During the past four months our system has not crashed once without our help. Except for performing periodic backups, and helping new users with the system, it has required little "management" work. It is not, however, bug free. In fact, compared with RSX11M V4.0, VMS seems to have a large number of small but nonetheless very annoying bugs and a few serious bugs. In RSX world, we had become very used to fixing bugs. If we found a bug, we either fixed the bug ourselves or we SPR'ed it and got a patch back in a month or so. Rarely did we invest time in finding 'work-arounds' unless the situation was critical. We would just wait and work on another project while the RSX developers fixed the problem.

The VMS world is different. The typical SPR response we receive is "Fixed in VMS V4.0". Serious bugs "may" get fixed in maintenance releases of VMS if they do not require major system changes. For example, we discovered that the VMS print spooler can not print files off of privately mounted volumes. Now, we consider this to be a serious bug, and if we had found such a bug on RSX, I am quite sure it would have gotten fixed rapidly. The VMS developer response to our SPR, however, was that that particular problem could not possibly be fixed before VMS V4.0 and would require creating new VMS access privileges. The realization that we are unlikely to get any serious bugs fixed, has forced us to invest time in developing semi-permanent "work-arounds" rather than wait for bugs to get fixed.

## RSX TO VMS MIGRATION EXPERIENCE

### Compatability, Similarity, and Incompatability Mode

When we started considering VAX migration, we were led to believe that almost all our programs would run directly on VMS with no conversion required. We just had to move the tasks over to VMS and run them in RSX Compatability mode. Soon we realized that the conversion process would be somewhat harder than we had anticipated. Our tasks are generally linked with a PLAS FCS resident library, and hence, can not possibly run on VMS. Moreover, most of our programs do file I/O, so to run on named directories, we had to relink them anyway. Since we had to relink, why not use VAX-FORTRAN to recompile them in native mode and save the expense of an additional F4P software licence for the VAX?

Doing this, we stumbled into a new 'GOTCHA' or bug almost daily which has significantly slowed our conversion efforts. For us compatability should mean more than just having a processor which can run a compatable subset of the PDP-11 instruction codes. It should also mean a simple FORTRAN program should compile (F4P on the PDP-11's, VAX FORTRAN on the VAX), link and run identically on both machines. While this is sometimes the case, often we encounter differences in behavior ranging from slight to major. Based on what we know now, FORTRAN programs now running on RSX systems seem to fall into three migration categories.

1. Compatible. Identical source code compiles, links, and runs on both systems.
2. Similar. Slight source code changes necessary to correctly run on the VAX but these changes can be back migrated onto RSX systems.
3. Incompatible. Major source code changes are necessary to run on the VAX and these changes will not work on PDP-11's.

### VAX and the AME user

VAX MCR users do not feel they are sitting at an RSX terminal. One of the first things a new VAX user (old RSX user) notices is that DCL is very wordy and VMS MCR bears little, if any, resemblance to the RSX task of the same name. The next thing the user notices is that it is very easy to get QUOTA EXCEEDED and ACCESS VIOLATION messages and not have any clue whatsoever as to which quota or type of access forced VMS to interfere with the work in progress.

Our first and simplest goal was to create commonly used command procedures to aid with system management and to provide users with functionality similar to that available in command procedures on our RSX11M system. Initially, we tried transferring the command procedures directly from RSX11M and using them by logging in with the MCR CLI. We had been a RSX11M V4.0 field test site for over a year, so our command procedures have embedded in them many features which are not in the VMS indirect file processor. Needless to say these command procedures did not run. In fact, RSX11M command procedures would cause the VMS MCR CLI to abort and log the user off.

We reported the problem with MCR aborting to the TSC, but they had never heard of the VMS MCR aborting and logging the user off. Since no solution seemed available, we abandoned the idea of using the MCR CLI and committed ourselves to rewriting our command procedures for DCL. While this option initially took longer to implement, it was a wise decision. The MCR CLI hardly seems to be a stable programing environment, and many of the features which make VMS attractive are not available to the MCR CLI. Moreover, switching between CLI's involves first logging off and then logging on the system onto a different CLI.

Rewriting the command procedures was a good learning experience. While it is possible to do almost everything from within the context of a VMS command procedure that is done by an

RSX indirect command file, accomplishing a given task generally requires more work since VMS has no permanently reserved special symbols which return status and device information the way RSX does. Also, debugging a command procedure is somewhat harder since one does not have a /-MC switch to inhibit the execution of commands.

Next we tried migrating some of our simple RSX11M tasks. All FORTRAN tasks on our RSX11M V4.0 system had been linked with our PLAS FCS resident library. While VMS will run tasks linked with resident libraries, it will not run tasks linked with V4.0 PLAS cluster libraries. This meant that we had to rebuild on the VAX any task we wished to migrate. In addition, tasks not linked with FCSRES on our RSX system and which did file I/O would not run without task building from named directory accounts. To further complicate migration, the V4.0 task builder is very different from the version of TKB in use on the VAX. For some of our work we will have to use the RSX11M V4.0 task builder which means migrating it to the VAX if V3.0 of VMS does not update to the V4.0 RSX11M task builder.

The RSX11M task builder runs more slowly under the AME than it does on our RSX system. This may in part be because the task builder on our RSX11M system is built with FCSRES. Whatever, the reason, task building a F4P program on VMS using a VAX 11/750 is almost a factor of two slower than TKB (/FCSRES) running on our RSX11M system. By contrast, the DCL LINK command will link an executable image up to 10 times faster than TKB.

We proceeded to bring over object module libraries from our RSX system and rebuilt a collection of Macro-11 tasks. Rebuilding the tasks allowed them to use the named directories and most tasks operated correctly. However, as soon as we started to migrate our data base entry and manipulation tasks (written using F4P) we discovered that the AME FCS had a fatal bug which we had not encountered on our RSX system. The VAX FCS would not let us do in-place updates of records within a sequential file whereas the RSX version would (provided that the total record length was not changed). Unfortunately, this was the first user application program we tried to migrate. The application had to be rewritten to use a different file access method.

The AME provides only a limited set of RSX executive services and often those services it does provide, are implemented in very different fashions and the code must be modified to run correctly. RSX V4.0 directives are not supported at all. Generally, these problems are not noticed with standalone tasks. However, tasks doing inter-task communication or spawning will have problems.

One of the most annoying features of the AME, it that it does not support tasks which use PLAS directives. Tasks using F4P virtual arrays or which use dynamic regions won't run. Since VMS supports the create and map region directive, it would seem logical that it could support the RSX PLAS directives in some fashion. Compatability mode tasks are also not able to link to the external page, or shared memory regions (in UNIBUS address space). This made it impossible to use the AME for migrating some tasks which did device control and which accessed a shared memory region connected to an image processor. Since native mode tasks can access shared UNIBUS memory regions transparently, this restriction seems artificial.

## RSX TO VMS MIGRATION USING DCL AND VAX FORTRAN

### Problems Migrating FORTRAN Programs

We had been a field test site for F77 and since VAX FORTRAN is supposedly a superset of F77, we expected minimal difficulties in migrating our FORTRAN programs to VMS. While the process has not been hard, neither has it been as straightforward and rapid as we would have liked. It turns out that VAX FORTRAN and F77 have many serious incompatabilities which were discovered buried in obscure places. Some of these incompatabilities are documented, some are not, and some are bugs.

The character data type is very different in VAX FORTRAN than in F77. VAX FORTRAN will not pass literal character strings to subroutines expecting LOGICAL*1 data. This means that we either must modify our programs so that they never pass literal character data to subroutines expecting LOGICAL*1 data, or we must have two version of the subroutine, one for passing LOGICAL*1 data and one for passing character data. Unfortunately, assembly language subroutines have no way of knowing how the data is being passed to them, or what the data type is supposed to be.

Our image analysis code had made extensive use of doing BLOCK I/O from FORTRAN. We would open files with BUFFERCOUNT=-1 and then do QIO's to the disk file. VAX FORTRAN does not support opening a file for BLOCK I/O with a BUFFERCOUNT=-1.

Because our image processor uses 8 bit data, we normally do math operations on LOGICAL*1 data. VAX FORTRAN generates an overflow error if a value greater than +127 or less than -128 is placed in a logical variable.

Error trapping in VAX FORTRAN works very differently in VAX FORTRAN than in F77. If an error is trapped by an ERR= transfer, the error status is never set. This prevents ERRTST from working correctly. This bug, we are told, can not possibly be fixed before VMS V4.0.

Control-Z trapping behavior is very different on VMS. To take terminal input on an RSX system after receiving a Contol-Z, you must close the terminal file and reopen it or just REWIND the terminal's LUN. VAX FORTRAN does not allow the REWIND of the LUN of a terminal.

VMS and RSX terminal drivers handle format control slightly differently. VMS insists on inserting a carriage-control, line-feed when a return terminates a FORTRAN READ. RSX only inserts a return. VAX FORTRAN does not allow a format statement to be encoded which has NULL carriage control. While such a difference seems insignificant, it became a serious problem for us when we discovered that tasks doing graphics to Tektronix terminals would not work correctly.

RSX TO VMS MIGRATION STATUS

Within the first four and a half months, we have accomplished the majority of our migration goals. The only programs we have not been able to migrate were those tasks which were heavily dependent on RSX11M inter-task communication or RSX11M V4.0 features (passing command lines to non-installed tasks, parent offspring tasking, etc.).

Since many of the tasks to be migrated used RSX11M executive service calls, rather than rewrite each task to use VMS executive service calls, we started creating a library of native mode VMS subroutines which emulate the RSX calls. This decision to provide RSX emulation subroutines, has the added benefit that a single source can compile, link and run on either the VAX or our RSX11M systems.

The VAX DECnet implementation also assisted our migration work. The report programs we migrated from our RSX system can transparently access any node in our network without any modification to the source code. Again, this allows us to maintain a single source for tasks to run on both our VAX and our PDP11's.

While migrating our image analysis work to the VAX required rewriting and converting a substantial number of subroutines from Macro-11 to Macro-32, the process was rather straightforward. On our PDP-11/45, the image analysis tasks had accessed the image processor through an 8 Kword bank of memory sitting on the UNIBUS. By creating a global section which overmapped those memory addresses, and providing the assembly language interfacing routines, the image analysis programs were rebuilt practically without further modification with the exception of those tasks doing fast disk I/O via block I/O and QIO's.

CONCLUSION

To date our migration from RSX to VMS experience has been a positive one. The process of moving our development and image analysis work over to the VAX seems much more like 'transmigration' than simple migration. For each program that is to be converted to run on the VAX, ways must be found to let the "soul" of the old RSX program live in the new body provided by VAX/VMS. Although we have experienced far more difficulties and incompatabilities than we expected in these efforts; the benefits of a more powerful and maintainable system seem to be well worth the costs. Ahead of us lies the task of routinely and productively using the VAX as a program development tool for RSX11M V4.0 systems .

# RT-11 to RSX-11M Conversion Experiences

Donald G. Seydel

P.O. Box 4765
Carmel, California 93921

Here are the principal perplexities I have experienced in conversion from RT-11 to RSX-11M. We are a small site (11/23) and BASIC-11 is important to us because it facilitates reporting in schemes and formats that are different than the original construction. We note the following differences between the RT-11 and RSX-11M versions of BASIC-11.

1. Files created by BASIC-11 on RSX-11M cannot be read by Fortran IV. The file header block will pick up a zero record size, rather than the correct size delared in the BASIC-11 dimension.

2. I have been unable to find a way to cause BASIC-11 to print directly to any device other than LP:, which means a lot of post-program PIPing to get the output where I want it.

3. COMPILE does not work.

4. If the last print line is concluded with a semi-colon to supress a line feed, the line will not execute. This has been noted at other sites. I recall from the last DECUS that the problem has been SPR'd several times without result.

5. Building a BASIC-11 with a shared library will not work as distributed. After much effort, I finally got copies of the three known patches for BASIC-11 published in the May 1980 Software Dispatch and found that one of the three addresses this problem. I have not had the time to go back and rebuild BASIC-11 to find out if this is true.

It appears that Digital is less than anxious to maintain BASIC-11 as a language usable on RSX-11M. The indications are:

1. It is not included as a layered language on the Autopatch kits, contrary to assurances I was given when I bought the Binary Program Update Service including Autopatch.

2. The May 1980 patches are not included in the February 1982 Software Dispatch Review recap for RSX-11M V4.0.

The Software Product Description for both RSX-11M V3.2 and RSX-11M V4.0 lists BASIC-11 as a supported language. I question whether this is really true.

Fortran IV moved over easily. I had to write string handling subroutines to emulate those found in the RT-11 SYSLIB. The main difference I encountered was the need to initialize a string by reading something into it before trying to INSERT into the string. This could lie with differences in RT-11's INSERT routine and my version. And RSX really dislikes byte arrays that end on odd boundaries, more so that RT-11. Finally, it took me a while to find that RSX wants to see "DISPOSE='PRINT'" to cause output to the printer.

FMS-11 forms libraries moved over without change. They can be editted with FED. I am told that once edited on RSX, they will not function on RT-11. At present, I do not receive any error messages from FMS (like when the library is not opened or a filed name does not exist) and I have not yet figured out what is happening.

# RSX-11M V4.0 KMSKIT Enhancements

James G. Downward

KMS Fusion, Inc.
P.O. Box 1567
Ann Arbor, Michigan 48104

## KMSKIT V4.0 Availability

KMSKIT V4.0 is now available for RSX-11M V4.0. It will be on the Spring 1982 DECUS RSX SIG tape. For those who just cannot wait and who are willing to make copies for others, I am willing to make a limited number of copies for distribution. Users interested in getting an early copy, must contact me prior to sending any tapes. Either write to the address above or call me at (313) 769-8500.

The kit includes only new and/or updated material for RSX-11M V4.0. This includes a System Accounting and Performance measurement package, tools for dynamically tuning your system, enhanced HELP processing, enhancements to SET and INStall, a multi-queue MCR command line despooler, and numerous other tasks. All modifications to DEC sources are done by auto-locating SLP patches. These have shown remarkable resiliency to surviving from release to release of RSX-11M over the past year.

Since the start of KMSKIT a number of years ago, it has grown to be quite a large package. It is scattered over a large number of UIC's ( [344,x] ) each with their own documentation files. Needless to say, ensuring that the documentation is self consistant throughout all the various UIC's is difficult. Every attempt has been made to document new features consistently and to provide adequate guidelines for incorporating the package. There probably have been slip ups and errors along the way. This is regretable but inevitable. The package, however, has been remarkably well tested by the RSX11M community. Hundred of installations are running all or part of the KMSKIT enhancements. If errors are found, please notify us.

Many of the enhancements, are implemented as SLP files for existing DEC sources. These SLP files have been written using ASCII locator strings and baring major surgery, should be useable even after DEC's SLP files have been applied. Be warned, however, these are only known to work using the distribution source files as input.

While every effort has been made to make different features in this kit independently selectable, please be warned that the only configuration ever tested at KMS is one in which all features are selected. Other combinations quite likely will work. That certainly was the intention.

A word about support. In the past, substantial effort has gone into providing phone support for users trying to put up KMSKIT. In most cases problems were resolved easily and were caused by not understanding or reading part of the documentation. With the release of KMSKIT V4.0, the V3.2 Kit is no longer supported. Also please note that some files are provided for convenience only and are not currently being used at KMS. Very little support (if any) will be offered on these.

Read through KMSGEN. It will attempt to do most all of what is required to put up the parts of the kit requiring a SYSGEN, Exec mods or privileged task modifications. The remainder of this document will try and describe the accounting, Executive and MCR enhancements in this kit

## ACCOUNTING ENHANCEMENTS

The RSX11M V4.0 Accounting Enhancements have been implemented to help manage the resources of a large, multi-user RSX11M system. The modifications to the system are not designed to make RSX11M into a full function time sharing system (although they come close), but rather have been implemented so a system manager can keep track of what we have found to be the system resources in greatest demand, namely disk block space, and terminal time. For large systems needing additional accounting capabilities, a comprehensive system accounting package is also included.

The goal of all modifications is to be completely compatable with a real time environment. The system overhead does not amount to more than several seconds per logon. The majority of the overhead exists at logoff when the account file is updated with the current disk block usage of the user. Since disk block accounting may be selected on a user by user basis, virtually no overhead need be incurred for very time critical applications.

An additional goal of the modifications, has been to increase the level of protection of a 'Multi-User Protection' System. In our environment a wide variety of users have access to the system, from secretaries using RUNOFF to lab technicians using BASIC, to tecnicians using F4P, to privleged systems programers. All of this must proceed concurrently with real time data collection. To insure this would be the case, two additional classes of RUN privleges are allowed (by setting bits in U.PRV1). Users may either be restricted to RUNning tasks only owned by them (they can't run system tasks like DSC or VFY or tasks built by other users), or they may be prohibited entirely from using the RUN command, and only be able to use tasks installed as MCR commands.

To allow protected processes to be implemented, by setting bits in U.PRV1, a task may be run automattically at logon, and/or the user may be logged on as slave. Slave logins are now a DEC option in the account file. By logging a user (an operator for example) on as privleged and slave, the indirect command file LOGON.CMD may be executed to exit in a totally protected manner. For example, a disk backup procedure which involves numerous privleged and dangerous functions (INS, INI, DSC, BRU) may be interactively executed with no chance that the operator will type in incorrect commands on his own and leave a scrambled backup.

To allow for detatched processes to go on after logoff and until RSX11M gets virtual terminal support, by setting a BIT in U.PRV1, no tasks are aborted at logoff. While this method clearly has limitations, it is better than nothing. Please note, however, that programs requiring to do file access and write files should not attempt to use this feature as the UIC gets zeroed when the terminal is logged off. A task allowed to run at logoff will keep the TI: the terminal. This feature is dangerous and should be used with caution.

Following the SPRING 79 DECUS meeting it became evident that there was substantial user sentiment for a more versitile accounting package for RSX11M. User's expressed need for tools to montior resourse and to provide performance measurement statistics for their systems. Also it was evident that very few users had exactly the same needs so that any system should be user extensible. Since RSX11M-PLUS was getting an accounting system some user's were annoyed that RXS11M was being overlooked. To alleviate this problem, this release of the KMS enhancements package contains user tailorable performance measuring tools.

The use of these tools is based on a user extensible accounting system. The interested user should read SYSLOG.DOC for the in depth description of the accounting package. Briefly, CPU time, NULL time, logons/logoffs, QIO's, connect time, disk block usage, illegal login attempts, POOL, memory, checkpoint files, Shuffler requests, checkpoint requests, pages spooled, and loader requests are all monitored and logged if desired. The user may select (on line) a veritable smorgesboard of options for selectively logging certain groups of data (or all) to a system log file. User by user CPU time and QIO activity can be monitored and logged. For use as a performance measurement tool, selective task accounting is available. The data is all logged to a file easily readable by a F4P task. Sufficient data is present to implement charge back accounting if so desired and a report program which does this is provided. When all options are implemented memory, POOL, or thrashing system bottlenecks can be discovered and monitored. Sufficient CPU statistics are gathered for those of us who occasionally must justify additional equipment, or systems. The system works very well and has been installed in hundreds of systems. It has no overhead unless it is started. It does, however, require a system generation and some minor exec mods. This, however, was felt to be a very low price to pay for the features offered.

RSX11M is very prone to thrash if the system is memory limited. RSX11M V4.0 has hooks to limit the number of runs the Shuffler can do within a certain period. The task SCH can be used to set the Shuffler run interval for tuning the system.

The MCR enhancements include a working single stream BATCH, a multiple stream batch-like Program Development Queue facility using the queue manager, enhancements to HELP file processing, a 2K procedure interpreter to replace ...AT. for simple command files, support for /NOLI in the indirect file processor task, enhancements to the SET command (LIBUIC, and TIMEOUT), enhancements to INS, and a super catchall task for RSX11M V4.0 which provides the user with a user extensible Console Command Language (CCL).

A single stream BATCH using a virtural terminal is provided. It BATCH uses the SPAWN directive and the Parent/Offspring support in RSX-11M. It does not use the Queue Manager. To implement BATCH two programs are provided, SUBMIT and BATCH. SUBMIT is used to pass the name of a command procedure file for BATCH to process. BATCH logs the user onto a virtual terminal, VT0: and executes the command file (using ...PIN) on the virtual terminal, and then logs the user off. One must be careful with BATCH not to have programs which prompt for input or which output excessivly. All output goes to a log file. Running RMDEMO from BATCH would quickly wipe out disk space. Likewise, if a task asks for input, null lines are output to disk file (sometimes a great many). (by the way, the account package does not log VT: connect time). The prime benifit which may be obtained from BATCH, is that it allows orderly program development to be done in an enviromment with limited memory and many terminals. Using BATCH only one copy of ...PIN., F4P or TKB need ever be present. This was found to be preferable to having four or more copies of each all buisily swapping in and out and never getting anywhere. To allow the users to monitor how BATCH is doing, BTQ is provided, to show the contents of the BATCH queue. The format of the output depends on whether or not the terminal is privleged (only privleged users get to look at passwords).

PIN is a procedure interpreter for RSX11M. It is used as a partial replacement for AT. If it is built with an FCS resident library it if over 4 times smaller than AT. (built with FCSRES). It is also faster. It supports a limited set of internal commands to control listing to TI: and to control treatment of non-fatal errors. MCRDIS has an optional patch to enhance the usefullness of PIN. The syntax "#CMDFIL" will execute a PIN cmdfile in much the same manner the "@" is used. Sample command files are provided to show how ...AT can be used transiently for creating build files suitable for use bY PIN, the compilers, and TKB.

Indirect has added a /NOLI switch. If used it has the same effect as starting out a command file with a .ENABLE QUIET. Also a new special symbol <ERRSEQ> is added so one can keep track of errors during automatic backup procedures.

The origional CCL has been significantly upgraded to provide additional error messages, exit with bad status if a command is not found, and to check for and execute the more common commands without having to scan the user and system CCL control files. It has been rewritten to provide comments and provide a more logical flow. Many new features have also been added including internal command tables. If RPOI$ support is supported on the target system, CCL will use it for sending commands to MCR and immediately exit. The user is urged to read CCL.DOC for a full list of features. MCR now supports a catchall task, and by installing CCL as ...CA., you have a super catchall task. A complete description of the benifits of CCL is described later. If the user intends to implement CCL, it is necessary to SYSGEN in the SPAWN directive (RPOI is optional).

## Utilities

Utilities written or modified in house which are of general use to the RSX11M community are supplied. The utilities include a

program to change a disk volume lable (and most other INI parameters) after initialization, an inter-user MAIL utility, disk fragmentation utility, a small program to list a file to TI:, a program to display which users (by name) are logged on the system (WHO), a program to display total user connect time and disk block usage, a program to provide a very fast total block listing of a directory, and task to display the contents of the BATCH queue, and a program to force MCR commands onto a remote terminal.

The origional CVL (Change Volume Lable) task was written by H. L. Coleman. Besides just changing the disk volume lable I have added support for /VI, /LRU, /WINDOW, /MAXEXT, /MXF switches on the command line (almost all the options INI has). This task has proved extremely useful not only in chaging the name of a disk, but also one can easily find out the name of a disk and all its significant characteristics (the /VI switch) with the disk on line. For example, there is no standard way (with the distributed system) of knowing what parameters a given disk has been initialized for (LRU, MAXEXT, MXF, ...). The mount command will display some information, but not all. Also it is sometimes very desireable to change volume characteristics from those when the disk was initialized. If a disk (say an RK05) becomes a single user disk, it saves pool to change the LRU count from the default of 3 to 2 or 1. At times it is possible to run out of room on a disk for creating more files even though there is ample room left. This situation can happen if the disk is initialized with a value for /MXF which is too low. It is possible to change the MXF count on the disk (upward but never downward) and add a few more files. The INI HOM command will do many of the same things that CVL will do. The one difference is that CVL will work on a mounted system device, wheras HOM requires the device to be dismounted.

Please note that the HOM command will do all CVL will do and more with the exception that it must be used on a dismounted disk. CVL can be used on a mounted disk and there are situations CVL can be used on a system disk whereas HOM cannot.

RMC is a FORTRAN task to force MCR command lines onto remote terminals using the SPAWN directive. It is very convenient. It only allows a privileged user to use this feature and will work with any number of terminals. It will first try and use the RPOI directive and if that fails it will try and use the SPAWN directive for sending the command. A task image and the Fortran source are supplied. RMC requires F77 to recompile.

A version of RMDEMO is supplied which is based on the old 3.1 RMDEMO. It is significantly smaller than the current version. Also it has the ability to provide a real time display of percent system utilization.

SCH is a task to change the Round Robin and Executive Swap intervals and a number of new system tuning parameters on line and to display their current status. Some of its features require that the KMS system accounting enhancments are SYSGENed in to your system. However, V4.0 of RSX11M has added global symbols in SYSCM which will let it change the Round Robin and Swap interval on line without a SYSGEN. SCH.HLP documents its use and command switches.

TCF is a task to send control characters to TI:. Some terminals use various escape and control sequences to perform various functions. TCF allows these to be sent conveniently from a command file. For example TCF 7 is bell and TCF 33,110,33,113,33,112,33,14 will erase the screen of either a VT52 or Tektronix graphic terminal.

## Key Features

### HELLO

1. HELLO will not allow multiple logons unless the user is privileged.

2. HELLO flags each active account with the name of the terminal using it, eg TTn.

3. HELLO notifies a user if any mail is waiting for him/her. It checks for MAIL on the "MAIL DIRECTORY" , LB1:[10,3]. LB1: is created most easily by making a global assignment of a physical disk to LB1: with VMR.

4. Stores the "Terminal Privlege Mask Word" in U.PRV1 of each terminal's UCB. For systems with only DL11 interfaces U.PRV1=U.CW3. Systems with multiplexers must modify the SYSGEN procedure to create an additional word in each terminal UCB, such that U.PRV1=U.LUIC-2.

5. If BIT 0 in U.PRV1 is set, then a user can not use the RUN command.

6. If BIT 1 in U.PRV1 is set, then a user may only run their own task or a task built with UIC=[377,377].

7. If BIT 2 in U.PRV1 is set, then a specified task (which must be installed) is run when HELLO exits. Also if BIT 2 is set, LB1:[1,2]LOGIN.TXT is not printed. If the requested task can not be run (not installed or busy) , a warning message is printed and the terminal is logged off. The name of the task to be run can be inserted into the account file (LB0:[0,0]RSX11.SYS) using the RESET utility. The various BITs in the terminal privlege mask word may also be set using RESET.

8. If BIT 3 of U.PRV1 is set the terminal has a QUIET login. No LOGIN.CMD files are executed. Bit 3 used to be used to log a terminal on as slaved. However, since DEC is now supporting that feature in ACNT, Bit 3's use is redefined.

9. If BIT 4 of U.PRV1 is set, do not abort non privleged tasks at logoff.

10. If BIT 5 of U.PRV1 is set the terminal is logged on with with NOBROADCAST attributes.

11. If BIT 6 of U.PRV1 is set, when the user logs off BYE sends an MCR command to ...UPD to update the user's account file with the number files and disk blocks in use on that account.

12. If BIT 7 of U.PRV1 is set, the user has group manager privleges. This user's login UIC's member number changes when SET /UIC used.

13. If BIT 8 of U.PRV1 is set, the user is 'non-privleged user manager'. By convention non-privleged accounts have UIC's >[310,x]. This user is allowed to change his LOGON UIC to any UIC as long as UIC>[310,x]. Otherwise user is treated identically with non-privleged user.

14. If BIT 14 of U.PRV1 is set, the user is allowed to dial in on remote lines.

15. If BIT 15 of U.PRV1 is set the terminal is logged on as privleged regardless of its UIC.

16. Warns the user if the number of disk blocks in use exceeds the number of disk blocks allocated for his use.

17. This release of HELLO accesses HELP information exactly the same as the DEC version with the exception of adding a 'MATCHALL' level identifier. If a '#' is found in column 1, it is considered to be a match for any qualifier at any level. Typically the 'MATCHALL' should be placed at the end of a HELP file. It may either be text or be a pointer (@FILESPEC) to a matchall HELP file. Each @HELPFILE may have its own 'MATCHALL'. We use the 'MATCHALL' to provide information as to what keywords are available in the file if the user requests a combination of keywords which would otherwise produce an 'UNKNOWN QUALIFIER' error message. For users with small system disks it is advantagious to sometimes be able to have help files on a disk other than the system disk. By default now HELP.HLP resides on LB1: not LB0:. LB1: is assigned globally to a physical disk either via VMR or in STARTUP.CMD.

18. If a "~" is placed in column 1 of a HELP file and the terminal is a CRT, a 2 second wait will occur when the "™" is encountered.This is useful if one has long help files and the terminal is a CRT.

19. HELP information display may be terminated at any time by typing ^Z. This feature requires unsolicited AST support at SYSGEN.

20. Information as to which HELP keywords are available at any level may be obtained by typing a "?" for that level of keyword. For example HELP topic ? would display all allowed keywords for 'topic'.

21. Included with the kit are substantial HELP files. Information on how to use most of the utilities on this kit is documented within the HELP files. The user of these HELP files may note that some information seems totally irrelevant. This is because no serious attempt has been made to weed out KMS specific files.

22. HELLO is modified so that if will not log logins on CO:. If the user wishes to log logins, the code to reenable this feature must be reenabled. The SLP file comments out a few lines and inserts a jump around a block of code. An examination of the SLPed file will show what needs to be done.

23. LOGIN.TXT is displayed only the first time a user logs in during a day unless LOGIN.TXT's creation date changes during the day. Hence, the display of LOGIN.TXT is independent of the form used for logging on.

24. As an aid in maintaining password security, the a single line login command of the form HELLO NAME/PASSWORD is overprinted to obliterate the password. CRT terminals are overprinted with spaces and hardcopy terminals with a line of XXXXX00000's.

### BYE

1. BYE updates the user's terminal connect time value at logoff unless the date has changed negatively or is greater than a value set at taskbuild time. Except for the last day of each month, this will provide for correctly accounting for user's logged on to a single terminal for more than one day, or during the late evening shift. The code required to handle (correctly) the case of rollover at the end of the month would also have to check for the case of end of the year. If a user requires end of the month rollover at midnight, the place to modify the SLP file should be obvious. If the date changes negatively, one must assume end of month rollover, compute the value for the last day in the previous month (accounting for February in leap year), and add the current day to the last day to find the effective current date prior to subtracting. If the rollover also occurs at the end of a year, the checks become even more elaborate. Please note that changing system time after users are logged on can produce bizzarr loggin time additions for the account file login in time.

2. BYE will not log off a privleged user if logons are disabled.

3. BYE passes a message to ...UPD to update a user's account file disk block usage after logoff.

4. BYE will print out the terminal connect time at logoff and if system accounting is installed, will also print out a summary of the user's CPU time and number of tasks run (just like M-PLUS).

### INS

1. Two classes of RUN privleges may be allocated. The checking of RUN privileges is done by INSTALL in the module INSHD.MAC. If BIT 0 of U.PRV1 is set the user is denied the use of the RUN taskname command. The user is however allowed to use tasks which are installed as MCR commands. If BIT 1 in U.PRV1 is set the user may use the RUN command, but only to run tasks owned by him or by UIC =[200,200] or [377,377]. Task ownership is determined by H.DUIC in the task's header. A user owns a task if H.DUIC = LOGON UIC, U.LUIC. In order to build a "MULTIUSER TASK" (one which all user's may use), a privleged user, or any user may use the UIC=[GGG,MMM] option to the taskbuilder. Restricting users to either running their own tasks or tasks built explicitly with the UIC= taskbuilder option, prevents users from using system tasks in a destructive manner. The primary reason for implementing this feature was to prevent users from accidently using DSC on a dismounted disk (which wasn't their's). DEC now has implemented run privilege protection in a slightly different way. In each file's protection mask, ([RWED,RWED.RWED.RWED]) , the "E" is taken to mean execute privilege is granted to either System, Group, Owner, or World.

2. The /PRM="text", switch syntax is allowed along with the DEC /CMD= switch to allow the passing of command lines to a task via the INS command. This additional command form is supplied on a onetime basis only to simplify changeover. The /PRM switch will not be supported in the future.

3. RSX11M has Executive support for LIBUIC and NETUIC. All that is necessary to use this feature is to make make a slight patch to the MCR SET parser and to modify INS to check both LIBUIC, SYSUIC, and NETUIC on an INS $xxx. After making the INS modification, the INS $FILENAME command will first search for the file on LIBUIC, SYSUIC, and then on NETUIC. This is a convenience feature for users who like to keep the privileged tasks on one UIC, network tasks on another and the unprivileged utilities on yet another UIC. It is also an RSX11M-PLUS feature.

4. If the system accounting package has been installed, INS will keep track of the number of times the user installs tasks (via RUN or INS).

MCR

An optional SLP modification to MCR implements several new features. The first modification implements two additinal user classes, the 'Group Manager', and the 'User Manager'. On a large RSX11M system it is often convenient to assign all programers working on a given project or class of projects, a common group number. To provide users protection from other users within their group, disks are initialized so the default file protection word is [RWED,RWED,R,R]. This protection mask prohibits group members from deleting or modifying each other's files. So that the onus of purging files no longer needed, of verifying all files are truncated or deleted if not needed, and of bookeeping does not fall entirely on the system manager, it is useful to appoint Group Leaders to perform these functions. A 'Group Leader' has BIT 7 in his user privlege mask word set ( U.PRV1). This bit grants him the privlege of changing his logon UIC along with his current UIC as long as his group number remains unchanged. A 'User Manager' has BIT 8 in the user privlege mask word set. If this bit is set, the user may change his logon UIC to any value greater than [Limit,X]. On our system the limit is [310,X] and User UIC's are selected to have group values >310.

In addition, the set command is modified to enable setting and display of LIBUIC via the set command (SET /LIBUIC=[GGG,MMM]). NETUIC support is already in SYSCM and SYSCM.SLP adds a support for LIBUIC. Finally, the SET /TIMEOUT command is added which will allow you to change the timeouts for devices. This is particularly useful for modifying timeout values on a per SCB basis for non-DEC terminals (ie Tektronix graphic terminals). This MCR set command will display or modify the timeout of a device DDn:. The command syntax is:

```
SET /TIMEOUT = DDn:       ! Display timeout
SET /TIMEOUT = DDn:M      ! Set timeout to M tics.
SET /NOTIMEOUT = DDn:     ! Disable device timeout
                          ! (set to 0 tics)
```

However, for this command to do any good for terminals and the FULL DUPLEX Driver, the timeout code in the FDT driver must be modified to use the timeout count (right now it is fixed, one value for all terminals). I have not done this yet. If someone has time to do this, please let me know.

Please note that to implement these modifications, you must select which terminal driver the system or the target system is using.

MCR...

All commands which would be normally rejected by MCR are now passed onto CCL for parsing. Also, the command form '#CMDFILE' starts PIN executing the command file in much the same way '@' starts up its command files. A simple way of looking at the logic of this choice is that '#' means direct execution of procedure files (MCR command lines) rather than indirect (interpretaton of many internal commands)execution. This change makes PIN as convenient to use as AT. for general program development.

UPDATE

1. Update will insert the the number of files and the number of disk blocks in use on the user's UIC and system device.

2. UPDATE can also provide the nonprivleged user with a very fast readout of the current file and block count. The required syntax is
   UPD DEV:[UIC].
   If just UPD is typed, UPD defaults the user's SY: and UIC. If UPD DEV: is typed, UPD defaults to the user's UIC. The total block and file count display provided by UPD is many times faster than the PIP /TB display.

## RESET

1. RESET is used to reset the terminal ACCOUNT BUSY FLAG in the account file. This should be done during STARTUP.CMD or in the event that a disk failure left an account flag set thus preventing the user from loging on.

2. RESET can reset the total terminal connect and CPU time and number for each logons of each user. This enables starting a new accounting period.

3. RESET can be used to set the user's disk block allocation, the user's Privlege Mask word, and an optional task to run after logon.

## ACCLOG

1. ACCLOG is used to provide a report of the account file status, showing the user's total connect time, total number of logins, disk block usage, disk block limit, Terminal Privlege Mask word, and optional task to run at login. In addition it displays the terminal any account whic is currently logged on to, and flags any account with a "*", if the user's disk block limit has been exceeded. User's wishing to display CPU time must be running the system accounting package and must have applied the SLP file to ACNT.MAC so that all account fields will be defined and useful (account number and ID).

## WHO

1. WHO is a mini-utility which provides a fast display of who is currently on the system and what terminal they are using. Type WHO to get the display. WHO also displays the user currently using BATCH (if any).

## SCH

1. SCH is a mini-utility which can display or modify the disk SWAP time and the ROUND ROBIN schedule interval (no SYSGEN is required). If the KMS EXEC mods have been applied it can also modify the round robin range and swap priority range. It allows the swap priority priority lower range to be alterted but not the higher range (that would have involved modifying two more EXEC modules). It will also set the interval between SHUFL runs. Finally, it will display all its setable parameters as well as a snapshot

of the system tuning parameters gathered by KMS system accounting. The format of the command line is

```
SCH /SW1.../SWn
```

where the switches are any of

```
  /DSP            Display the Round Robin and Swaping interval.
  /RBN:n          Set the Round Robin interval to  n  decimal
tics.
  /RBL:n          Set the Round Robin low priority range to  n.
  /RBH:n          Set the Round Robin high priority range to  n.
  /SWP:n          Set the Swap interval to  n  decimal tics.
  /SWL:n          Set the lower priority for swapping to  n.
  /MSF:n          Set the maximum # of SHF runs/swap interval.
```

An error message will result if an attempt is made to set the Swap interval
to less than the Round Robin interval. The /DSP, /RBN and /SWP switches will
work on any RSX11M V4.0 system.

## MAIL

1. The utility MAIL (...MAI), provides an inter user mail facility. At logon a user is notified if he has "mail" waiting to be read. He may retrieve his mail by typing "MAIL ME". This utility is supplied courtesy of Mark Johnson.

2. A user may send mail to another user by typing "MAIL TO NAME", where NAME is the last name of the user (or his account file name) to whom mail is to be sent. If a valid username is found in the account file, MAIL prompts for a message. The user can type in line after line of a message until done. The message is terminated using a control-Z (^Z). MAIL is stored on LB1:[10,3].

3. Documentation on using MAIL is found in MAIL.RNO.

## BATCH

Two batch or batch like facilities are provided, BATCH and PDQ. BATCH provides a single stream development queue using a virtual terminal. PDQ provides a multi-stream Program Development Queue using the Queue Manager and the issuing terminal.

To use BATCH, the tasks BATCH and SUBMIT must be built. The file BATCH.HLP (a HELP file) provides information on its use. Before building BATCH, it is necessary to edit BATCH.MAC and insert the physical device to which the log file will be written. Log files are currently being writen to DMO:. BATCH does not know the SY: of the user submitting a command file (the UIC and password use up too much space in the send buffer.) so all users using BATCH must have their files on the selected disk.

The only way around this problem is a rewrite and logic change of SUBMIT/BATCH. If the instalation selects to have all LOG files spooled, this problem would go away, since BATCH could write all LOG files to a single UIC (say [10,1]) on a previously specified disk and spool them to a line printer. We chose not to do this as we have a number of remote terminals. It would be very inconvenient for the users to have to come retrieve their output from the line printer. It is also possible to configure BATCH so it writes its output directly to a terminal rather than the logfile. This would slow down the execution of any command file producing a lot of output (output to a disk file is faster than to a terminal) and would leave the terminal active until output is finished.

The program BTQ is provided so that both the system manager and the general user can monitor the number of command files in the queue to BATCH. This task is privleged and must be rebuilt each sysgen. The report format to a privleged user will supply the origional command (including passwords) submitted by the user. The non privleged user will only get a list of command files not yet looked at by batch. In the event that a user submitts a command file which hangs BATCH up in an endless loop, the system manager can user BTQ to provide a listing of commands queued up but not yet processed. Then BATCH be aborted, and the list of files resubmitted for processing. Please note. BATCH is now not dependent on special versions of HELLO or BYE!! SPAWNING solves all previous problems!

PDQ (Program Development Queue) is a multi-queue despooler for MCR commands. The name of the command file for PDQ to execute is submitted to the queue manager via the CCL EXECute command. When a queue becomes free the commands start executing at the user's terminal. This makes for an orderly program development cycle on a crowded system, and causes escentially no delay on an uncrowded system. MCRQUER.RNO contains further documentation on PDQ.

SYSLOG, ACC, TASKLOG, LOGTSK

As an aid in measuring system performance, SYSLOG, ACC, TASKLOG, and LOGTSK are provided. SYSLOG provides a resource accounting and performance measurement tool for RSX11M V4.0. ACC is the MCR interface task to the system logging task SYSLOG. SYSLOG inserts intercept code into the EXEC to monitor QIO's, null time, CPU time, memory usage, pool usage, etc. Task accounting is implemented via TSKLOG which requires syslog to be active also. The task accounting is primarily designed to be a performance

measurement tool. When requested TSKLOG will request a list of up to 25 tasks to be entered. These tasks will have their QIO's and CPU time monitiored. The interested user is urged to read SYSLOG.DOC to get additional information on system accounting capabilities provided by these tasks. Data sent to LOGTSK by SYSLOG, TSKLOG, HELLO, BYE, and UPD is written out on LB:[1,4]SYSLOG.DAT. SYSLOG.RNO defines the data formats (sequential, unformatted, variable length records) and data packet contents.

CVL -- CHANGE VOLUME LABLE

CVL is a privileged (build each sysgen) external MCR COMMAND. It's function is to display the disk volume name, many disk initialization parameters, and the volume creation date, and to optionally change most of these parameters. The command form is

```
        CVL DDn:LABEL/SW1/SW2/SW3...
```

where SW1,....SWN ARE ANY OF

| | |
|---|---|
| /VI | Display volume information but don't change anything. |
| /MAXEXT =NN | Change the defalut file extension to the new value NN. |
| /WINDOW =NN | Change the default number of windows to NN. |
| /LRU =NN | Change the default LRU the volume will be mounted with. |
| /MXF =NN | Change the maximum number of files allowed on a volume.<br>(Only increases upward are allowed.)<br>(NN is decimal in all cases). |

The LABLE specified in the command becomes the new volume name.

AT., MCR, and CCL

One of the biggest problems experienced by the RSX11M user is lack of pool space. It is convenient to have all utilities installed, but the more tasks installed the smaller pool becomes. Also one is limited to only three character names for tasks installed as MCR commands. CCL together with the AT. and INS enhancements eliminate or alleviate almost all of these problems. In reality the amount of code changed in MCR, AT., and, INS is rather small, however the effect is large. AT. now will look on SY:[1,5] if it can't find the command file the first time. This enables a system to maintain a catalog of commenly used command files equally accessible to all users. Since parameters can be passed on an indirect command line (@commandfile, param1, param2,.....) a CCL command of the form

```
        BUILD MYPROGRAM,LIST,BUILD,NOMAP
```

could assemble, list, and taskbuild a F4P program via a cataloged command file in a method completely transparent to the user (Nothing appears at the terminal if the /NOLI switch is used until the command file is done.). Commonly used system tasks not requiring long command lines, may be made to appear to the user to be installed where in reality they are being implemented via a

```
          INS filespec/CMD="MCR COMMAND LINE".
```

This can result in a substantial POOL saving.

A new special symbol <ERRSEQ> is added so that the indirect command file has access to the current number of errors on a system. Routine tape backups are often done (via BRU) via indirect command file procedures. By keeping track of <ERRSEQ> before and after a backup, one can keep track of the number of recoverable errors on a tape. If the number gets too large, the tape should be scrapped.

The reader is urged to read CCL.DOC for detailed information on CCL.

## INCORPORATING THE PACKAGE

Transfer the files on [344,x], to the UIC's of your choice, and type @KMSGEN . Please note that KMSGEN.CMD will not work if the files are on the system UIC. Please note that it is preferable to leave the files on their distribution UIC's since this is the configuration which was tested. These files have only been implemented on a mapped system. Modifications to the build files should make it possible to use these modifications and programs on an unmapped system, but the only way to find out for sure is to try. Most enhancements are conditionalized on parameters which get placed into RSXMC.MAC during the modified SYSGEN. If one really knows what one is doing, some files can be assembled by hand to incorporate certain features without doing a SYSGEN. However, you are on your own on this.

************** WARNING **************

As distributed, the simple accounting package can be incorporated without a SYSGEN only if the users system has no terminal multiplexers (ie only DL11'S), has not selected DECNET support, and has not selected multiple CLI support. These programs all assume that U.CW3 can be used as the TERMINAL PRIVLEGE MASK WORD. This assumption is OK FOR DL-11'S. However, multiplexers also use U.CW3. The solution is to add one word to each terminal UCB(U.PRV1 = U.LUIC-2). To simplify the process, SYSGEN.SLP SGNTT.SLP, SGNEXEC.SLP, and SGNPER.SLP are provided. By applying the SLP files before starting a SYSGEN, a terminal UCB will be produced which has the requied additional data structure in front of U.LUIC. Note, that multi-user protection is required for the package to work. If a SYSGEN is to be done, one should apply SYSGEN.SLP, SGNTT.SLP, SGNEXEC.SLP, SGNPER.SLP and TDSCH.SLP and SYSCM.SLP, do a SYSGEN and then build this packages features as time and opportunity allows. Please note that additional problems can occur if DECnet support is elected and the data base for HT: must also be modified. This is done with the SLP file HTTAB.SLP.

KMSGEN.CMD will build the KMS utilities. It will attempt to SLP the required DEC sources and request that you complete the process by doing a sysgen. Note, INStall may be installed with VMR or more conservatively one may proceed as follows.

```
                    INS INS;-1/TASK=...TNS
                    REM INS
                    TNS INS
```

Now one can try out the new copy of INS with no risk to the system. Using the new version of INS will enable RUN privileges to be assigned on a user by user basis (via the PRIVILEGE MASK word in U.PRV1 of the terminal's UCB). However, modifying INS is not required to make the modifications to HELLO, and BYE work. ACCOUNT.HLP provides documentation as to what the various bit setting in the PRIVILEGE MASK WORD will accomplish. The choices are 1) the user may not run any uninstalled task, 2) the user may not run task not owned by him (H.DUIC <>LOGIN UIC) , 3) an optional task is requested to run as HELLO exits, 4) the terminal is logged on a privileged regardless of UIC, 5) A non-privleged user is to have his account updated to reflect the number of disk blocks and files he is using, 6) the terminal is to be logged on as slave, and 7) users may be granted dial in privileges.

Remember for CCL, MAIL and HELLO to work the logical device LB1: should be assigned to a disk via a global assignment in STARTUP.CMD. Also it is advisable to add a line of code to reset all account busy flags when the system reboots. The tasks ACCLOG AND RESET do not have to be rebuilt each SYSGEN. UPDATE is the task' ...UPD which updates the user's disk block usage when he logs off. This feature is selectable on a user by user basis by setting a bit in the privlege mask word. It makes no sense to apply this to a privleged user who has files on more than one UIC and/or on more than one disk. The task ...UPD is privleged and must be rebuilt each SYSGEN. The user will probably want to change the title line in ACCLOG.MAC. To reflect the name of the user's institution rather than KMS FUSION. To do so EDI ACCLOG.MAC and PF TITLE1:.

Documentation on the utilities is in ACCOUNT.HLP which is in HELP file format. It can either be incorporated into the system's HELP.HLP or used by itself. Other .HLP files are also provided (in V4.0 format). To use the MAIL utility, UIC [10,3] must exist on logical device LB1:.

If just the HELLO, and BYE modifications are made, [0,0]RSX11.SYS will start keeping a running tally of total connect time by each user. Only privleged users are allowed to be logged onto more than one terminal at a time. If the system is improperly shut down ( a crash?) users may still be logged on. Upon rebooting these users will not be able to log on unless privleged. At this point the system manager must install RESET as ...RSE, type RSE FLAGS. This will zero all the account file busy flags. This step may be incorporated as a normal part of STARTUP.CMD.

******* RESTRICTION

A problem exists because SHUTUP does not log off the terminal from which it is run. If SHUTUP is always run from the console device for the system (the terminal the system boots with) then no problems exist. However, if SHUTUP is run from some other terminal, then the account busy flag (of the form TTn) for that account is not reset. For a multiuser protection system, the the system boots with the console device logged on so as a normal procedure in the STARTUP command file, the console (boot) terminal is logged off at the end of the command file. This resets the proper account busy flag ONLY if SHUTUP was run from that terminal. Otherwise it notices that no flag of the form TT0 exists in the account file, and prints a warning message. However, an account file busy flag exists still under another account. If someone then goes to use the non console device terminal from which SHUTUPrun the night before, he is logged on and now the account file has two account busy flags of the form TTn in the account file under different accounts. When the user goes to log off, the first account found with an TTn is reset. This may or may not be the user who is loging off. If it is not the user, the next time he tries to log on, HELLO will notice that his account busy flag is still set, will inform him (if he is non privleged) that multiple logons are illegal, and prevent him loging on.

This is a permanent restriction. Since the system only knows the terminal number and account number when a user logs off, there is no unique way of finding the user's account if more than one account busy flag per given terminal gets set in the account file.

Two solutions exist for this problem. First, one can always run SHUTUP from the console device. Second, in the STARTUP.CMD file, one can install RESET and reset all flags (RSE FLAGS). In practice this solution works well. A STARTUP.CMD file might look like

```
INS $RESET
RSET FLAGS
     "
     "
     "
BYE
```

By including BYE at the end of STARTUP.CMD, the terminal (which is logged on during STARTUP.CMD) is logged off. However, because the account busy flags have been reset, BYE can't find which account to compute connect time for and prints a warning message before loging the terminal off. (BYE -- STILL LOGGED ONTO FIRST TERMINAL?). This message is normal and shows that indeed RESET and BYE are reading and writing the account file correctly.

If difficulties, or bugs are found, please contact me

James G. Downward
KMS Fusion, Inc.
3941 Research Park Drive
Ann Arbor, Mich 48104
313-769-8500

and I will try and be of help and fix bugs as time allows. I would appreciate it if suggestions for improvements are passed on to me.

# Process Control Questionnaire

Mark King
Process Control Working Group

In the last few months, the PROCESS CONTROL working group has conducted a simple questionnaire amongst the people that have placed their names on the working group's mailing list. The purpose of this questionnaire is to establish a base of information on the people actually doing process control/data acquistion applications. Approximately 56 questionnaires were mailed with 24 responses being used in this summary.

1. Average number of years programming DEC equipment is 6.

2. When asked about any problems with DEC software or hardware, most of the respondees stated that the software documentation was difficult to use or information was lacking in detail. However, there were a few people that said they had no problems with either software or hardware.

3. All the people responding said that they would be willing to share hints and techniques in using process equipment.

4. Only three people said that theyd be able to attend the Spring Symposium. A few of the ones that could not attend stated that the cost could not be justified to their company.

5. A general remarks question was asked concerning DEC equipment. Here are a few of the comments in edited form.

   - DEC does not indicate how to optimize RSX11M for process control or data acquisition application. It really seems to be slanted toward program development or interactive programming.

   - The hardware is good but the software to use it properly is late or non-existant.

   - Using DEC I/O in process control is expensive in time and resources.

   - FORTRAN is not good for a user interface. Use of a PASCAL compiler would be nice.

This questionnaire was not really an indepth survey. At the DECUS symposium a suggestion has been made to provide a new questionnaire to the RSX community to really obtain solid indicators of the process control/data acquisition users needs. After June 1st, I will be working a new job in Maryland. My address will be Mark W. King, EMC Controls, Inc., P.O. Box 242 Cockeysville, Md. 21030. Phone: (301) 667-4800.

## Process Control Questionnaire

This questionnaire is directed at the RSX11M, M+ and VMS systems that perform process control and/or data acquisition functions. The main concern is with the DIGITAL MDC product line (Manufacturing, Distribution and Control).

Return this questionnaire to Process Control Survey, c/o Multi-Tasker, DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752. The results will be tabulated by the working group and published in the Multi-Tasker.

Circle the appropriate answer at the end of each question

1. Would you attend product panel sessions presented by the Manufacturing, Distribution and Control product line (MDC)? Y N

2. Would you attend product panel sessions presented by the Computer Special Systems group (CSS)? Y N

3. Would you attend question and answer sessions with MDC? Y N

4. Would you attend question and answer sessions with CSS? Y N

5. Would you be interested in forming a data acquistion/process control SIG that encompased RSX11M, RSX11S, RSX11M+, VMS? Y N

6. Would you be willing to work within the DA/PC SIG doing any of the following (circle all that apply):

   a. Chair sessions at DECUS
   b. Present topic papers
   c. Panel presentations on selected subjects
   d. Other

7. Would you like to see the RSX/IAS SIG sponser 1/2 day of DA/PC related sessions? Y N

8. Would you prefer question and answer sessions to product panel discussions with CSS or MDC? Q/A PP

9. Would you like to see more process interface equipment on exhibit at future symposiums? Y N

10. As a member of the RSX/IAS SIG, are you more interested in (circle all that apply):

    a. Applications under RSX/IAS.
    b. Operating system enhancements and new features
    c. Seeing RSX11M stablized at version 4.1
    d. Migration to VMS
    e. Continuation of RSX11M as a mainline DEC product

11. Did you join the RSX/IAS SIG simply because your site uses that operating system? Y N

12. Do you feel that your needs and wants from DECUS are (circle one):

    a. Greater than 75% operating system related information and activities
    b. Less than 75% but greater 50% related to the operating system
    c. 25% to 50% related to the operating system
    d. Less than 25% related to the operating system

13. With respect to the proceding question, please check the other areas that are not related to the operating system directly.

    ____ networking

    ____ field service

    ____ cpu and/or memory

    ____ dual system configurations

    ____ training

    ____ communication hardware

    ____ bus switching

    ____ peripheral switching

    ____ environment and power

    ____ problem solutions

14. Have you ever used the DECUS SIG tapes? Y N

15. Do you have access to the DECUS SIG tapes? (i.e. compatible medium) Y N

16. Do you belong to a local users group (LUG)? Y N

17. Do you plan to attend the next fall or spring symposium (circle one)?

    a. Fall
    b. Spring
    c. Both
    d. Neither
    e. Unsure

18. Would you attend an evening "BIRDS OF A FEATHER" sessions for the process control working group? Y N

19. Does your site use (circle all that apply):

a. RSX11M
b. RSX11S
c. RSX11M Plus
d. IAS
e. VMS
f. other

20. In five years, will your site be using (circle all that apply):

a. RSX11M
b. RSX11S
c. RSX11M Plus
d. IAS
e. VMS
f. other

21. Is your site planning to migrate from (circle all that apply):

a. RSX11M to RSX11M Plus
b. RSX11M to VMS
c. RSX11M PLUS to VMS
d. Other

Comments:

Please return to Process Control Survey, c/o Multi-Tasker, DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752.

DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MR2-3/E55
MARLBORO, MASSACHUSETTS 01752

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 129
NORTHBORO, MA
01532

```
INSTALLATION     - 138550
DAVID J. STRAIT
APPLIED DYNAMICS INTERNATIONAL
3800 STONE SCHOOL ROAD
ANN ARBOR  MI   48104
```

**MOVING OR REPLACING A DELEGATE?**

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

( ) Change of Address
( ) Delegate Replacement

DECUS Membership No.: ____________________

Name: ____________________

Company: ____________________

Address: ____________________

____________________

State/Country: ____________________

Zip/Postal Code: ____________________

Mail to: DECUS - ATT: Membership
One Iron Way, MR2-3
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here. Include name of installation, company, university, etc.