THE MULTI-TASKER

Volume 14, Number 5 May 1981

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

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

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

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

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

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

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

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

TABLE OF CONTENTS

Columns

From the Editor. 112
Hints and Things 112
Core Tuning a RSX-11M System
Macro-11 Errors
Software Performance Reports 114
F4P Include Files
EDT Taskbuild
Macro-11 Assembler and Lower Case
F4P ANSI Tape I/O
Chris Doran Collection
Help Yourself. 117
DX11-B UNIBUS to IBM Channel Interface
XY11 Device Driver
Three Dimensional Drawing
MCR Functions for Octal/Hex
VT30 Device Driver
IGL Graphics Package

Articles

Floating-Support for ODT 119
The Best of ICR. 121
Table Driven DCL Command Language. 123
Running IAS V3.1 on a LSI-11/23. 127

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

FROM THE EDITOR

The highlight of this issue is that is got published at all. I have been extremely busy and am sorry for the delay in getting this to the DECUS staff for mailing. However, things will not improve for the next issue because of the Miami Symposium. So expect the June issue to be delayed a few weeks and I will try to get back on track in July.

Ralph Stamerjohn
Multi-Tasker Editor

HINTS AND THINGS

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

The following article is from Wilfried Sander, Finnigan MAT, 28 Bremen 14, W. Germany.

CURE TUNING A RSX11M SYSTEM

Several notes have been made in the past in the Multitasker, how system performance can be improved. Indeed there are many ways to improve throughput, but size reduction of tasks is not in any case the right way, because core size is not the only aspect. I think, CORE*TIME is a better indicator for system throughput.

The system comes with task sizes that are not suitable in some cases. Some are far too big, others are too small! Candidates that come too small and should be INCreased are in any case BIGMAC abd BIGTKB. Others come with proper sizes like CRF and DMP. Some others are too big, among others there are INS, RMDEMO, MOU, and DMO.

Let me first show what happens with the tasks that come too small. In systems which have the extend task directive, BIGMAC and BIGTKB increase their size at runtime. But this takes time. So it is the better way to find out, what the typical MAC and TKB size for their own application is and to install TKB and MAC with this size. The time reduction on our system was about 40%, when I installed BIGMAC at a typical assembly close beyond the final size instead of the minimum size.

A much more dramatic example of what can be done wrong is F4P. It comes with a small size and does not allocate more workspace when needed. So we have to install it properly. By the way this is very well documented in the F4P INSTALLATION MANUAL, but one has to find out, which INCrement is best. So we took a typical program size and configured the F4P corresponding to this. Our increment is 4096 and the time savings are more than 70% compared with and increment of 1024.. These times show how important the correct INCrement is. DEC has learned, by the way. The F4P V2.5 came along with an increment of 2048., the V3.0 comes with 3840., which is close to the value we found best for a wide range of program sizes.

Because most of the tasks are heavily overlaid, it is in any case best to install the most used ones from the fastest disk that is available. In my performance measurings I found a total time reduction of over 60%, when I installed the F4P with an INCrement of 4096 from a fixed head disk compared with the default from a RK05 disk.

Another improvement could be made, if the system came with a FCSRES. The former RSX11D had it already and I really don't know why the 11M doesn't come with one too. So we have to help outselves (look on the DECUS tapes).

And now to the candidates that come too big. Why do they come with an improper size? Does DEC want to force us to a 11/44? Most installations have only a core of 124K and cannot bear large task sizes. RMDEMO, INS, MOU, DMO, ACS, LOA, UNL, PRT can be reduced with no risk.

But we have to be careful when reducing the size of others! Many tasks do not increase their workspace and spend significantly more time when installed too, small, or they run out of workspace. These are among others CRF, DMP, CMP, LBR. They shouldn't be installed under 8K.

The following suggestion is from the editor's own collection of tricks.

MACRO-11 ERRORS

Sometimes, even I have errors occur when assembling a program. In order to save paper, I use the following TECO command on the listing file. This command will find and list all lines marked in error. This is because the assembler always puts the error code in column 1 of the listing. You can find these errors by searching for <CR><LF>non-blank character. Note, the command also finds all subtitles.

```
        >TEC file.LST
        *<N
        ^E^A$;V>$$
```

SOFTWARE PERFORMANCE REPORTS

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

The following SPR on RSX-11M V3.2 F4P was submitted by David M. Kristol, (Mass. Computer Associates, 26 Princess St., Wakefield, MA., 01880).

F4P fails with a FATAL 05 error when the logical end of file of an INCLUDE file exactly lands on a block boundary and the last line is another INCLUDE file reference.

The following SPR on RSX-11M V3.2 EDTBLD.CMD was submitted by Dennis T. Cook, (Fischer & Porter Company, Dept. 436 - Systems Center, County Line Road, Warminster, Pa. 18974).

Running SYSGEN3 from an RL distribution, gives the following TKB error when building EDT:

```
        TKB -- *FATAL* module SAVRG NOT IN LIBRARY
```

The module is really in the library. Moving [1,20] EDTBLD.ODL, EDT.OLB AND [1,24] EDTBLD.CMD to the system disk and executing EDTBLD causes the problem to disappear.

The following SPR on MAC was submitted by Wilfried Wober (Technical University of Vienna, Karlsplatz 13, A-1040 Vienna, Austria). Wilfried also submitted the response he received from Digital.

STATEMENT: MAC incorrectly converts lower case characters to upper case when storing the body of a macro or repeat block.

RESPONSE: To correct the problem, find the address and block number for the symbol "PROMPT" in the MAC.TSK task builer map and alter the contents of location PROMPT+154 from 52 to 42 as follows, using ZAP:

```
>ZAP
ZAP>MAC.TSK
_<B>:<A>;0R
_0,154/
_<B>:0,154/52
_42
_X
```

In the above dialogue,<B> and <A> are the block number and address respectively of "PROMPT".

The following SPR on ANSI Tapes and FORTRAN was submitted by Peter Bendall (EMBL c/o DESY, Notkestrabs 85, 200 Hamburg 52, Germany). Peter also submitted his response from Digital.

STATEMENT: An F4P program creates a file with unformatted writes, which is to be read with a second program using unformatted reads.

When attempting to read the file from ANSI magnetic tape, the FORTRAN Object Time System terminates the program with error 37, Inconsistent Record Length.

DMP reveals that the MT: records contain both ANSI and Files-11 type record length indicators. The ANSI byte count is thereby 2 higher than it should be, with the extra two bytes appearing before the data.

IBM objects to the record length not being 32 bit binary.

RESPONSE: When a file is created on ANSI magnetic tape, certain fields in the header are filled in with appropriate values. The field "Record Length" located in HDR2 is required to contain the maximum record length for a file containing "D" format variable length records.

FCS does not permit specification of a maximum allowable record length as RMS does, and thus this field contains the maximum possible length -- the block size. However, FCS does maintain a field which contains the maximum actual (not permissible) record length for the file. This is unknown at the time a magtape file is created.

When the file is opened for read, the MTAACP returns the maximum allowable record length to the issuing program. Unfortunately, the FORTRAN OTS validates this length against the MAXBUF parameter specified at task-build time and rejects the open request unless there is enough buffer space available to read what it believes would be the largest record in the file.

For the time being, you may avoid the problem by specifying a MAXBUF parameter equal to the magtape block size being used. We will investigate other methods of solving this problem for a future release. We believe that we should not return a maximum record size to the issuing program when it is not known, but will have to assess the impact of such a change on FCS, RMS, FORTRAN, COBOL, BASIC, etc.

The records are, indeed, recorded in the correct format. The ANSI record size is the size of the record as far as the File System is concerned. For unformatted data records the FORTRAN OTS maintains its own information in the first two bytes of the record. This format is documented in the Fortran IV Plus User's Guide. Since this record format is a private, DEC-FORTRAN, record format, it is not expected to be compatible with other systems. If you wish to interchange with IBM, you should write the records as formatted records.

The following SPR's were submitted by Chris Doran, SIRA Institute Ltd., South Hill, Chislehurst, Kent BR7 5EH).

Since DEC seems to publish only a small fraction of the SPR's to which they reply, I am sending you resümes of some of those I have submitted, in hope that they may save some head-scratching.

1. Error in FLX Patch 5.1.7.1 M (Lone line feeds) Change the location of label LF$PAT from .BLK.+450 to .BLK.+440, otherwise the line-feeds are left in, and everthing comes out doublespaced. The new .POB file checksum is 3114.

2. PAT always exists with warning status (0) even if there are no errors. This is corrected by:

```
        .TITLE PATBL
        .IDENT /02A/
;
;MODIFICATIONS:
;
;       02A - - PROVIDE EXIT STATUS WORD THAT IS NOT CLEARED
;
        .PSECT PATCHA

$EXSTS::BLKW    1       ; EXIT STATUS WORD THAT IS NOT CLEARED

        .END
```

   Checksums are: PATBL.OBJ=62702, PATBL.POB=4624.

3. CDA won't accept CDA as its output filename. (Correction in next release).

4. DMO crashes if an attempt is made to dismount, giving a volume name, a magtape which isn't mounted. (Correction in next release).

5. PIP's /CD switch cannot be used on ANSI magtape transfers. If you do use it, some very funny things happen to the creation date/time in the disk file headers. (/CD will be prohibited for magtape in next release).

6. The RSX-11M-PLUS macros VRCD$S, VRCS$S, and VRCX$S will cause an odd address trap if the send/receive buffer size is defaulted to 13. This is because of an incorrect BL=13. in each macro header which generates MOV 13.,-(SP)! To correct them, extract the three macros from LB: 1,1

RSXMAC.SML and patch each BL=13. to BL=#13.

7. When including a user-supplied $DELCK in LPP it is necessary to add it to the ante-penultimate line of LPPBLD.ODL, which becomes:

```
                    PD:      .FCTR   LPP/LB:PRTDN:DONE:$DELCK
```

   Note also that $DELCK may not modify R1, and possibly other registers.

8. BASIC-11 writes non-ANSI format magtapes, even if rebuilt with ANSI support. I am told this will take "too much work to fix".

Problems with BASIC-11 don't seem to get much attention. I am waiting for answers to:

1. Typing an illegal function name (e.g. FNM1$(X)) may crash BASIC-11.

2. If the last PRINT to a file ends with a semicolon, that line will not appear when the file is listed.

HELP YOURSELF

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

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

THIS MONTH'S QUESTIONS

DX11-B UNIBUS TO IBM CHANNEL INTERFACE

I am looking for an RSX-11M Driver for a DX11-B interface attached to a PDP-11/55 and an IBM system/370 selector or block-multiplexor sub-channel. The interface is to be used to transfer files between the 370 and the 11/55 and possibly from other PDP-11's connected to the 11/55 by means of DECNET. If anyone knows of such software, I would be pleased to hear from them.

John Seddon, Computation Centre, National Research Council, Montreal Road, Ottawa, Ontario. K1A 0R6 Canada. Phone (613) 993-3238.

XY11 DEVICE DRIVER

We have a PDP 11/34A with a DEC XY11-C plotter interface. We need an RSX11M driver so we can drive our Calcomp plotter using the XY11-C.

Philip Brewin, The University of Aston, College House, Gosta Green, Birmingham, England, B4 7ET.

THREE DIMENSIONAL DRAWING

I am looking for software which will enable draughtsmen in a drawing office to build up three-dimensional shapes using a graphics terminal and to produce drawings with dimension lines and annotation.

Graham Sharples, Royal Aircraft Establishment, Computation Division, Instrumentation & Trials Department, Farnborough Hants GU14 6TD.

MCR FUNCTIONS FOR OCTAL/HEX

Where can I find MCR-commands that give me the following functions:

- o Octal/hex/decimal arithmetic
- o Conversions to and from octal/word, octal/byte, hex/word, ASCII, RAD50

We have RSX-11/M V3.1.

Jacob Bagge, Elverket, S-631 86 ESKILSTUNA, Sweden Phone +46-1610 2536.

VT30 DRIVER

We have a controller VT30C and bought the software (sources) for handler and editor from DEC 1978. Now in 1980, we tried to install the software and found that it doesn't work. Unfortunately, the period of guarantee is expired. Has anyone a controller VT30C or VT30D in use with DEC-software under RSX-11D or RSX-11M?

Reinhard S. Goering, Deutsche Transalpine Oelleitung GmbH, Vogelweideplatz 11, D-8000 Munchen80 (German Federal Republic)

IGL GRAPHICS PACKAGE

I would like to hear from anybody about the IGL graphics package from Tekronix. Any user comments about installation and/or operating problems would be appreciated.

C.C. Bryant, Altec Lansing, 1515 South Manchester Avenue, P.O. Box 3113, Anaheim, California, 92803. Phone (714) 774-2900.

FLOATING POINT SUPPORT FOR ODT

Chris Doran
SIRA

I have extended V3.2's ODT to include support for programs using the FP11A floating-point processor. Five different examination and modification modes are implemented:

1. 1-word floating-point numbers, as used by immediate mode instructions (LDF #^F1.5,F0 etc.).

2. 2-word numbers (single precision).

3. 4-word numbers (double precision).

4. Access exponent bits as an unbiased octal byte (LDEXP/STEXP compatible).

5. Examine/write FPP status, or examine exception code and address (FEC and FEA).

Two additional command characters are defined for the extended ODT: '[' opens an address in floating-point mode, and ']' closes it with the option to enter a new value typed in floating-point notation (modes 1,2 and 3), or octal (modes 4 and 5). The command syntax is:

```
r,a[m
```

where r,a may specify an address in the usual way, or may be omitted to use the current address. It may be replaced by $0-$5 to open the floating-point registers, $S to open the FPP status word, or $W to examine FEC and FEA. Except for the last two cases, m is an additional character which is required to completely define the floating-point mode. Legal characters are /, E, or D, selecting 1-,2-,or 4-word numbers respectively, and \ to access the exponent.

Hence if F5 contains PI, location 1234 contains 1,23456789123456789D23, relocation register 3 contains 1000, and there is an LDF #^-1.5,F0 at address 2000, then the following terminal dialogue might appear:

```
_1234[E 0.12345678E24                ;Open 1234 in "E" mode
_3,234[D 0.12345678912345678D24      ;Open 1234 in "D" mode
_$5[E 0.31415926E1                   ;Open F0    in "E" mode
_2002[/ -0.15000000                  ;Open 2002 in "/" mode
_$S[ 000200                          ;Open FPP status
_$W[ DZ: 3,76                        ;Open FEC, FEA
```

where the bottom line indicates that the last floating-point exception was a divide by zero at address 1076. FEC codes 0-14 are reported by mnemonics: OK=0, OP=2, DZ=4, IC=6, OF=8, UF=10, UV=12, MT=14; anything else is printed in octal.

To prevent exceptions occurring in ODT a special test is made for undefined FPP bit patterns, and the word "Undefined" appears on open in such cases.

After printing a floating-point value ODT returns to command level for another instruction. It expects numbers to be in octal so to change an FPP location it is necessary to tell it when a floating-point number is to be supplied. So an open location may be closed by a command of the form:

```
]vt
```

where v is the new value and t is one of the terminators: CR (close only), LF (close, open next) or ^ (close, open previous). If v is omitted the contents of the current location remain unchanged; if it is given, a decimal floating-point value is accepted for modes 1,2, or 3, or an octal number for modes 4 or 5. A floating-point number may be entered in fixed-point notation in which case it is stored in 1, 2, or 4 words depending on the last mode used. (The inital default is 2 words). Alternatively, it may be given in scientific notation, in which case the exponent symbol, /, E or D, determines the current and subsequent modes. Opening the exponent only does not change a previously-defined number length. If the number terminator is LF or ^, the last /, E or D sets the number of bytes by which ODT's current address is updated.

Note that the] is mandatory if a location is to be changed correctly, even when the FPP status or an exponent is being modified by typing an octal value, as ODT must be able to distinguish this from a normal command. There is no FPP instruction to modify FEC and FEA, so $W v will be rejected. Note also that the next or previous location will be displayed in floating mode only if explicitly closed in floating mode, so to display a series of consecutively-stored numbers several LF's must be typed. The internal workings of ODT make it difficult to implement floating list or search options.

The floating-point extension is written as a PAT correction file, and thus easily added to the distributed ODT. It uses no EIS instructions, but has only been tested on the FP11A processor. Since it consists of 700 lines of code it is impracticable to list it here; source copies as listings or paper tape can be requested from me, or I will copy onto supplied media (600' magtape, RK05, or RL01).

Chris Doran
Sira Institute Limited
South Hill
Chislehurst
Kent, BR7 5EH
England

THE BEST OF ICR

Robert Stodola
Institute for Cancer Research

We at the Institute For Cancer Research are distributing a tape (The Best of ICR), containing a few useful programs. Some of them are IAS specific, but many would be at home on an RSX system. The tape is available to anyone who sends a 2400 foot tape with a return mailer to the address given below. In addition, these programs will be on the spring Decus RSX Sig tape.

A short description of each program follows:

1. LIST - LIST is a super file lister. Lines from a file may be displayed in several modes, including a screen at time. You can search for character strings and write portions of files to disk. In addition, LIST can use SRD to list multiple files, making it easy to do such chores as searching through a group of files for a particular string. Useable on RSX or IAS.

2. FDT - Number one on the RSX and IAS menus was a Fortran symbolic debugger. Here it is! RSX or IAS.

3. CPL - CPL (compile) figures out what compiler to use and compiles your program, but only if the creation date of the source is later than that of the object file. In addition, it handles multi-pass compilers, including RATFOR, Swedish and NBS Pasca, BASIC+2, and Whitesmith's C (as well as F4P and MACRO, of course). RSX or IAS.

4. SRDCMD - SRDCMD is a flexible command line generator which uses SRD to select files, then writes command lines constructed from specified text and the device, uic, name, extension, and version fields of the selected file names. RSX or IAS.

5. SRD - An enhanced version of SRD including more powerful file string matching, selection by file revision date, and storage allocation summary. RSX or IAS.

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.

7. BRU - You will need BRU to read the RSX Sig tape; this version includes the modifications to the TU16 device handler which are necessary to use BRU. Note: This is a prerelease of the IAS V3.1 BRU, supplied to us solely for the purpose of reading the RSX Sig Tape. It is most unofficial.

8. BURSTF - BURSTF bursts Fortran subroutines, functions, main programs, blocks data from a file and writes them to individual files. RSX or IAS.

9. RESEQ - RESEQ resequences the statement labels in a Fortran program. RSX or IAS.

10. 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.

11. XEQ - XEQ runs programs. It maintains a global common area which contains, for each user, a list of directories to search when XEQ tries to run a program. Never again will you have to type XX0: [100,222] or whatever 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.

12. 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.

13. GAME - GAME uses SCHEDULE to restrict game playing to specified times. IAS and RSX (see SCHEDULE).

14. RUNNL - RUNNL is a companion program to SCHEDULE and GAME. It will run a task on the pseudo device NL. IAS only.

15. RATFOR - an enhanced and debugged version of the Software Tools Group's Ratfor compiler, this version features structured Fortran output and specification of FORMATs within READ, WRITE, ENCODE, and DECODE statements. RSX or IAS.

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 Computer Services; 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.

To get a copy of the tape, send a 2400 foot tape with return mailer to: Robert K. Stodola, Computer Center, The Institute for Cancer Research, 7701 Burholme Ave., Philadephia, Pa. 19111.

A TABLE DRIVEN DCL COMMAND LANGUAGE INTERPRETER

Freek de Kruijf

Computing Centre
Delft University of Technology
P.O.Box 354
NL - 2600 AJ Delft
The Netherlands.

We have developed a table driven command language interpreter (CLI) with the following features:

- an absolute control of the user environment
- a table driven CLI
- a hierarchical table structure
- a subtable structure
- a subset fo the DCL syntax
- an integrated help facility
- a multi task command with conditional task execution
- a small task
- an efficient table search

The design has been based on the CCL task of Richard Kirkman and the KMS enhancement package of James G. Downward. We feel that our work should be integrated with the work of Andy Rubel (Multi-tasker August 1980). The task can be installed either as the catch-all task '...CA.' or as '...DCL', but in the catch-all mode only part of the features is available.

1.0 ABSOLUTE CONTROL OF THE USER ENVIRONMENT

The absolute control of the user environment becomes available via additions to the KMS enhancements. These additions are:

1. A new bit in the userprivilege mask is defined as the DCL bit. This bit forces the HELLO task to change the dispatcher task of that terminal from 'MCR..." to 'DCL...'. The U.CLI entry in the terminal UCB has been used to implement this feature. The system must support multi CLI's. The HELLO task will also start the CLI task, either in cleanup mode or in normal mode.

2. If the update bit has been set in the userprivilege mask BYE will check, using UPDATE, whether the user has exceeded the allowed number of disk blocks, if the check fails BYE will not logoff the user and exits with an error status code. On logoff BYE will reset the U.CLI entry in the terminal UCB.

The additions force absolute control of the user environment, because the MCR commands are no longer available to the user.

2.0 CLI TASK

In the catch-all mode the CLI processes the command and exits. In the other modes the CLI reads a command from the terminal, processes this command according to the current table(s), reads a new command etc. until an exit command defined in a table will be reached. After typing C and a command, this command will arrive at the dispatcher task 'DCL...'. This task will send the command by one or more send directives to the CLI task. A receive AST in the CLI makes it possible to process this command asynchronously from the normal command processing. Only the STOP command will be accepted which will abort the current running command if this command has been defined in the table to be abortable.

3.0 HIERARCHICAL TABLE STRUCTURE

At the start of the CLI a begintable must be established. In cleanup mode the begintable is in the file LB:[1,5]CLEANUP.DCL. In the normal mode the begintable is either the usertable, the grouptable or the systemtable, LB:[1,5]ggguuu.DCL, LB:[1,5]ggg000.DCL and LB:[1,5]SYSDCL.DCL respectively, whichever will be present first. Here ggg is the octal groupnumber and uuu is the octal usernumber. The begintable may contain as the last record the filename of a continuation table. A maximum of three concatenated tables is possible.

4.0 SUBCOMMAND TABLE

As the result of a command the begintable may be replaced by a special table belonging to that command. The commands defined in this new table and the possible continuation table(s) may be seen as subcommands of the invoking command. An exit subcommand forces the return to the user-,group- or systemtable. The CLEANUP table can be seen as a special subcommand table.

5.0 SUBSET DCL COMMAND SYNTAX

The CLI recognizes the following elements in a command

```
        ccc[/qqq[:vvv]] [ppp] [rrr]
```

Where ccc is the command name, qqq is a qualifier name, vvv is a qualifier value, ppp is parameter-group-] and rrr is parameter-group-2. The square brackets specify optional syntax. These elememts may have any length and may contain any character except spaces and tabs, ccc may not contain square brackets or a slash, and qqq may not contain square brackets or a colon. The elements ppp, rrr and vvv are character strings which can be inserted in skeleton MCR commands in the table. Each command ccc or a combination ccc/qqq should have an separate entry in one of the tables. The entry specified defines also the number of required parameters. If a parameter is missing the CLI will prompt the user for it with a prompt message from the table.

6.0 INTEGRATED HELP FACILITY

By designing a questionmark '?' entry in the table and a proper structure of help files, the user may get information on the available commands or a specified command by typing a questionmark optionally followed by a commandname. The questionmark may also be given as an answer to the prompt for a parameter. In this case a help command given in the table together with the prompt message will be used to produce the information on the parameter.

7.0 MULTI-TASK COMMAND

The syntax of a command entry in the table permits the invocation of more than one task. The invocation of a task can be made dependent on the exit status value of a previously invoked task. Also messages in the tables may conditionally be given to the user.

8.0 SMALL TASK

The CLI does not have advanced features in the table structure. Advanced features may be implemented using the indirect commandfile processor task. Moreover, DEC will come with DCL, so we decided to a moderate effort in implementing this task. For the time being this task will suit our needs. It has the advantage of a small task, which has been designed with separated readonly and readwrite modules. The readonly part can be installed as a resident library. For each user only the data part will be duplicated.

9.0 EFFICIENT TABLE SEARCH

Tables in files means alot of file I/O to find the command in the table, which makes the command processing not efficient. In catch-all mode hardly anything can be done to improve this, in our mode a first improvement is to save the file-id of the table files. A better improvement has been found in what we call a memory resident file. A memory resident file is located in a dynamic region in memory large enough to hold the file. The first two words of the file-id of the file are used as the name of the region. This region will be shared by other users. The first CLI task which wants to access the file should create the region. The other CLI tasks have to do an attach to this region and, if they want to access the table, perform window mapping.

Bruce C. Wright
Duke University Medical Center
Durham, N. C. 27710
(919)-684-6804

Running IAS V3.1 on an LSI-11/23

The Clinical Computing Facility at Duke University Medical Center has had an interest for quite some time in running IAS on an LSI-11/23. The necesary hardware has only recently become available; much of it involved items from various manufacturers. The machine is now fully operational and running with the following hardware:

1. An LSI-11/23 processor with the floating point chip and memory management

2. A 124K-word MOS memory card from Dataram Corporation

3. Two RM02-compatible 67-megabyte disks from Braegen Corporation

4. A controller for the Braegen RM02-compatible disks from Emulex Corporation

5. Two RX02 floppy disk drives from Digital Equipment Corporation

6. A backplane, power supply and line clock from ADAC Corporation

7. A BDV-11 bootstrap and bus terminator from DEC

8. A DZV-11 four-line terminal interface from DEC

9. A DLV-11 console terminal interface from DEC

We are currently running IAS V3.1 on the system. DEC does not officially support IAS on an LSI-11/23, although there are only a few fairly simple modifications which are necessary to get it to work. Note that the patches and techniques mentioned below have been developed and tested under IAS V3.1 and may or may not work with previous releases of IAS or RSX-11D. Also, the fixes may or may not work with the new PDP-11/24, although with the new support for the PDP-11/44 which is in IAS V3.1 there is good reason to believe that it could run on an 11/24 with minimal to no changes. Note also that any installation contemplating running IAS on an LSI-11/23 should come to some understanding with DEC about licensing IAS for the system.

Probably the most important thing to bear in mind about running IAS on an LSI-11/23 is that the IAS system expects to have a real-time clock running. The Adac power supply provides a real-time clock, but it is controlled from a switch on the console rather than from the CPU like most of the real-time clocks on most of the PDP-11 family. Without its real-time clock, there are a number of system components which may not work: Sysgen phases 1 and 2; various device drivers (which use the clock for timeout of requests); the heuristic scheduler; and of course the time of day status. When IAS is booted, the real-time clock should be turned on as soon as possible or the system may mysteriously "go away" without warning. Usually it can be brought back by simply turning on the clock.

The LSI-11 family bootstraps expect a bootstrap block to start with a NOP instruction (octal 240). The IAS bootstrap block starts with a branch instruction, followed by "SYS" in Radix-50. For us, this wasn't much of a problem, since we were booting a disk not supported by the LSI-11 family bootstrap; we just read in block 0 of the disk and jump to it. This technique would work on the DEC-supported devices; however it is not difficult to patch the bootstrap block on the disk:

```
SCI> SET BOOTSTRAP SY0:[11,17]IAS
SCI> DISMOUNT SY0:
SCI> MOUNT/UNL/OVERRIDE:VOLUME SY0: xxx
SCI> MCR ZAP
ZAP>SY0:[0,0]INDEXF.SYS/AB
_0/
000:000000/ 000505
_240

000:000002/ 075273
_504
_X
SCI> DISMOUNT SY0:
SCI> MOUNT/OVERRIDE:VOLUME SY0: xxx
```

The effect of this patch is to change the bootstrap block (block 0) to start with a NOP instruction followed by a branch. Note that it is not possible to change this in the appropriate RxxxBOOT.TSK module since then BOO (for example) will refuse to recognize the system bootstrap as a legitimate bootstrap and will not write the bootstrap block. Either the change must be made everywhere (in all of the bootstraps, BOO, SAV, and Sysgen Phase 1) or it must be patched after the fact in the index file. Naturally while the index file is patched the system must be quiescent.

One of the quirks of the LSI-11/23 is that it does not follow the usual convention of the PDP-11 family for odd address trap detection. On the LSI-11/23, a trap through location 4 will only occur when a non-existant address is referenced. On most PDP-11's this will also occur when an odd address is referenced in word mode. On the LSI-11/23, the effect of referencing an odd address in word mode is that the address is rounded down and refers to the previous even address. For most correctly executing user and system programs, this has no effect; however, the SAV task does a special check for a hardware ECO which must be installed in a PDP-11/40 in order for IAS (or any other system which uses virtual memory mapping) to run. The check does not work properly on an LSI-11/23, and will cause severe corruption when the system is booted. The fix is quite simple:

```
        SCI> MCR ZAP
        ZAP>SY0:[11,1]SAV.TSK
        _3:3642/
        003:003642/ 012707
        _5737
        _X
```

The effect of this patch is to change a MOV #3,PC instruction into a TST @#3 instruction. On the LSI-11/23, the MOV #3,PC instruction causes it to branch to location 2; on most PDP-11's it causes an odd address trap. Since the only purpose of the instruction is to try to cause an odd address trap, the TST instruction should be used. On an LSI-11/23, it will not cause an odd address trap but will fall through normally and reach the interrupt vector routine anyway. The fact that the stack doesn't end up at the same address on an LSI-11/23 as it does on another system is irrelevant since the stack pointer is reset a few instructions down anyway (no RTI instruction is ever done, nor is there any dependence on the contents of the stack until after the stack pointer is reset).

An incompatibility which might affect some user software (although it does not affect the IAS system software) is the LSI-11/23's handling of the RESET instruction. On most PDP-11's, the execution of a RESET instruction in user mode will act as a NOP instruction; on the LSI-11/23, it will be treated as an illegal instruction. The HALT instruction is also treated as an illegal instruction on the LSI-11/23, rather than causing a trap to 4 as it does on most PDP-11's. User mode software normally should not use these instructions, but there is a possibility of some software somewhere which could be run in either user mode or kernel mode and which would count on not getting traps from a RESET instruction or getting a trap to 4 from a HALT instruction. This is probably not a very important incompatibility.

The DZV-11 is a four-line terminal multiplexer which looks very similar to the DZ-11 except for the fact that the DZ-11 has eight lines to the DZV-11's four (the DZV-11 has sufficient bits in its control registers to support eight lines but the bits are ignored). The IAS Terminal driver, TT64, will correctly handle

the DZV-11 if it is declared in CONFIG.MAC to be a DZ-11 but only four of the eight lines normally allowed to a DZ-11 are declared. The various DLV-11's will also work in the terminal handler since they look exactly like a DL-11.

Altogether, we are quite pleased with the few number of problems encountered in running IAS on an LSI-11/23; most of the problems which we encountered were hardware problems incurred mostly because of the variety of manufacturers and some local hardware which has been used on the system; very few of our problems have been software-related, and most of those were finally traced to the problems enumerated above. Because many hardware problems can be related to software problems, and because even the above problems can occur in a number of different disguises, we would strongly reccommend that anyone considering running IAS on an LSI-11/23 have the IAS sources available; if not on site, then available for perusal at the local DEC office or at another nearby site.