THE mini-tasker

DECUS
RT-11 SIG NEWSLETTER

JANUARY 1981 VOL. 7 NO. 1

Contributions to the newsletter should be sent to:

Ken Demers
MS-48
United Technologies Research Center
East Hartford, Connecticut 06108
203 727-7241

Other communications can be sent to:

John T. Rasted
JTR Associates
58 Rasted Lane
Meriden,Conn. 06450
203 634-1632

or

RT-11 SIG
C/O DECUS
One Iron Way
MR2-3/E55
Marlboro,Mass. 01752
617 481-9511 Ext. 4141

FROM THE EDITOR

Please contact me if you were at the 'RT-11 Can Do It' session at San Diego and have the session taped. I would like to include parts of the session in the next newsletter.

Thankyou,

Ken

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.

USER INPUT

Chester & Lois Wilson
Phone (0745911)-201

71 Galatea St,
Charleville 4470,
Australia.
17-Oct-80

Dear Ken,

I have had some trouble getting much interest in Runoff from your side of the Pacific, but the efforts of Bob Denny (if I read his signature correctly) in the last Mini-Tasker have prompted me to ask if you could please add this letter to your publication.

The version of RUNOFF (M01) which I submitted to DECUS some months ago has prompted a wish list (see below), and a couple of fairly easily correctable bugs have been found by David O'Connor (of Buffalo). Here they are.

PROBLEM 1: No Hyphenation

This is a source-level patch in HYPHEN.MAC. Calls to ALPH and ALPH2 are tested on return with BEQ and BNE instructions, instead of BCS and BCC instructions. Here is the TECO command string for correction of this, with dollar signs being escapes, and "^A" being control-A.

```
.R TECO
*ebHYPHEN.MAC$$
*<nPC,ALPH$;
lcc
::fsNE$CS$"u::fsEQ$CC$"u
^A? Error in HYPHEN.MAC
^A2v''>$$
*ex$$
```

PROBLEM 2: /P repeats Headers

This is an error of omission on my part. The routine in RNRT11.MAC for getting a line from the user's terminal (TTC4N) is missing a RETURN statement at the end, with the result that execution falls straight through into TTC3, sending the header (if one is present) to the output file before it actually should get there.

The TECO instructions for correcting this are as follows:

```
.R TECO
*ebRNRT11.MAC$$
*nTTC4N::$n.PAGE$0L$$
*<tab>RETURN
$$
*ex$$
```

(The <tab> is a single tab character.)

".special <delimiter><characters><delimiter>" with control characters being permissible within this string. It may be useful particularly for screen use to have strings predefined (prefix file) for different terminals and called merely by ".special <name>" where <name> has been defined in a ".spcdef /string/" as per .special.

6. It would be useful to have a ".tell <text>" to type information to the user during processing. Similarly a ".wait" - for example, to allow changing daisy wheels or golf balls on a printer.

7. Some ability to cope with proportional spacing printers, for example by reading in a table of divisions for each character.

8. An ability to underline blank spaces.

9. Ability to put page numbers on either (or alternating) sides of the page, or in the middle at the bottom.

10. (This is a crazy one to see what you think): A relatively simple form of conditional processing could be implemented now that it is possible to have multiple file input. Consider:

I am aiming to collect assorted ideas and bugs, and re-release to DECUS in about 6 to 12 months. The original aim had been to leave the source code completely alone so as to maintain RSX compatability, but after the HYPHEN source bug was detected I reckon extensions are definitely in order, keeping the sources so that they may be passed across to anyone from RSTS/RSX who wants to upgrade!

Current gripes/bugs: (to solicit those of EVERYONE interested please!)

1. Assorted blank pages at end of output (probably will tackle by holding output in RNRT11 and ignoring blank pages there)

2. .noheader leaves a blank line at the top of a page.

3. .autoparagraph causes a bombout if there are blank lines at the end of the input file. It also causes a break performs an implicit .paragraph, which is most undesirable if it is to be included in a prefix .RNO file. For compatability it may be necessary to have a ".setap" command, which just sets up the parameters as per .ap but doesn't have the additional effect.

4. .literal fails to include blank lines. I feel this is a darned nuisance.

5. A mechanism should be included to allow addition of special characters (eg for setting up LA120 or Diablo printers), with no recording of them from the point of view of horizontal or vertical counting. It could take the format of:

```
.SETTRUE PARAM1 (in prefix file)
...
.IFTRUE PARAM1
<runoff commands etc>
.IFEND (?.ENDC PARAM)
(similarly .IFFALSE)
```

I would appreciate any input you have re all this. Runoff is getting a fair pounding this side of the Pacific, and the better we can make it thr more use to everyone.

Any more bugs anyone can find will be gratefully appreciated.

Regards,

[Chester Wilson]

THE MEDICAL COLLEGE OF PENNSYLVANIA INCORPORATED MARCH 11, 1850

The Medical College of Pennsylvania

3300 Henry Avenue, Philadelphia, PA 19129
215-842-6588

DEPARTMENT OF RADIOLOGY
Radiation Physics and Safety Office

Wayne A. Meyers, M.S., Director
Michael G. Stambaugh, M.S.
John G. Wolodzko, B.S.

6 November, 1980

Ken Demers
MS-48
United Technologies Research Centre
East Hartford, Connecticut 06108

Dear Mr Demers,

The following note is intended to spare some grief for anyone who may initialize a disc, by design or otherwise, only to find that their backup volume is unusable.

As you may be aware, section 8.2.12.5 of the System User's Guide for RT11 version 4 describes the DUP option switch /D to "uninitialize" a volume immediately following an initialization. But, on the occasions I've had to employ this technique, it has been completely ineffectual, the only result being the error message: "DUP-F-INVALID RESTORE DATA-DK1:". This means that the device Initialize/ Restore Data Area at locations 204-251 of the home block does not contain a duplicate copy of the device directory (for an unexplained reason) which existed before the initialization.

Now, contrary to the statement accompanying the error message, all files and derectory entries existing prior to the initialization are not lost. Initialization does not delete the original device directory. The directory segment link words are altered, however, in addition to other changes, to indicate that no segments are in use. By employing the save image patch utility, SIPP, described in chapter 22 of the RT11 version 4 System Reference Manual, it is possible to patch the device directory segment link words, as described in section 9.1.6.2 of the Software Support Manual. A careful review of chapter 9 makes the process quite clear and simple.

A final note of caution: Apparently the first few files on a disc can be lost during an initialization, but most of the user's files, if not all of them, will be listed in the directory subsequent to the above procedure. Other parameters, such as the total number of blocks, or free blocks, on the device will be incorrect, but these factors do not interfere with the transfer of files to another disc.

Yours sincerely,

John G Wolodko

BIOMEDICAL ENGINEERING UNIT
FACULTY OF MEDICINE
McINTYRE MEDICAL SCIENCES BUILDING
3655 DRUMMOND STREET, MONTREAL, QUE., CANADA H3G 1Y6
McGILL UNIVERSITY

October 7, 1980

Mr. Ken Demers,
MS-48,
United Technologies Research
Center,
East Hartford, CT
USA 06108.

Dear Mr. Demers:

Some years ago I submitted descriptions to the 'Mini-Tasker' and the 'Multi-Tasker' of two RT-11 related systems which have since found popular acceptance in the community. These were:

RDCL: A system which allows an RT-11 computer to transfer files to and from an RSX-11M system and also provides 'virtual terminal' support from the RT system to the 11M system;

RTV: An implementation of RT-11 V02B which runs as a task under RSX-11M.

Development of the 'RDCL' system has continued since that time. To date, there are over 50 installations worldwide which run RDCL. Some installations use RDCL to support a large number of 'core-only' LSI-11's running RT-11 for production, dedicated-type applications. Others use RDCL as a means of augmenting the mass storage of their disk-based RT-11 machines by using 'virtual disk' space on the 11M machine. Most of the users have reported that the system runs very well, is efficient, and constitutes a viable alternative to DECNET. Users who would like more information on the latest release can receive a full documentation set (free of charge) by writing to me.

RTV was originally the brainchild of Mr. Bill Clogher, then a DEC systems programmer, who, in his own words "developed a prototype one evening after a party" in order to convince DEC management that such a product was easily implemented and that a market would exist for it. Bill handed over the system to me at a DECUS meeting in 1976 after it became evident that management was not interested. I did a bit of development on it, and was using it in-house for a while in addition to distributing it to interested users.

Because the original design of RTV was 'quick and dirty', it proved to be an impossible job to upgrade it when RT-11 was upgraded from V02B to V03. However, user interest was still (and is even now) high and so I continued to distribute it. Several unsuccessful attempts were made at U.S. DECUS meetings to convince DEC that a market does exist for the product, but to little avail. We got vague hints and 'don't quote me' promises that funding for it existed and that it was just a matter of time before it became available. I have finally given up.

My purpose in writing this is twofold:

a) I wish to inform interested users that I will no longer distribute RTV; and

b) I wish to encourage interested users to send purchase orders to DEC for "RTV V01.0" as a method of demonstrating that there is a market for such a product. I would suggest that they make purchases conditional upon 90 days delivery.

Judging by the number of enquiries wich I receive for 'RTV', I'm fairly convinced that this strategy may achieve a useful purpose. It would be interesting to get an 'official' DEC reply to this letter published in some future issue of the Newsletter.

Thank you for the space.

Yours truly,

Gary Bernstein,
Manager, Computer Systems,
Biomedical Engineering Unit.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ DECUSCOPE : DIGITAL EQUIPMENT COMPUTER USERS SOCIETY$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

APPLICATION NOTE :   RT 11 (FORTRAN IV SOURCES.)
******************   APPROXIMATION FOR DIGITAL COMPUTERS.

A SHORT PROGRAM TO ESTIMATE THE VARIANCE RATIO LIMIT F AT A GIVEN LEVEL OF PROBABILITY AND NUMBERS OF DEGREES OF FREEDOM

****

BY DANIEL GUINIER

LABORATOIRE DE PHYSIOLOGIE COMPAREE DES REGULATIONS
GROUPE DE LABORATOIRES DU CNRS DE STASBOURG-CRONENBOURG
23 RUE DU LOESS
B.P.20 CR
67037 STRASBOURG CEDEX   FRANCE

INTRODUCTION :
****

IT IS CONVENIENT TO HAVE AN APPROXIMATION OF THE LIMIT OF THE VARIANCE RATIO F AT A GIVEN LEVEL OF PROBABILITY AND NUMBERS OF DEGREES OF FREEDOM AVOID TO REFERRING TO TABLES WHICH ARE OFTEN IMPRECISE. MANY METHODS HAVE BEEN DEVELOPPED BY DIFFERENT AUTHORS, THE PRESENT METHOD IS SHORTER AND FASTER.

WE HAVE MADE MORE THAN 4000 TESTS TO COMPARE THE RESULTS OF OUR METHOD WITH THOSE OF ONE OF THE CLASSICAL METHODS.

REFERENCES :
****

ABRAMOVITZ M.,STEGUN L.A. : HANDBOOK OF MATHEMATICAL FUNCTIONS
(DOVER PUBLI ,INC. NEW-YORK EQUAT 26-2-23).
HASTINGS C. (1955) : APPROXIMATIONS FOR DIGITAL COMPUTERS
(PRINCETON UNIV PRESS NEW-JERSEY).
IMSL (1978) : SUBROUTINE MDFI
JASPEN N. (1965) : THE CALCULATION OF PROBABILITIES CORRESPONDING TO THE VALUES OF Z, T, F AND CHI2.
(EDUC PSYCOL MEASUREMENT VOL.25 PP.877-880).
UNIVAC : SUBROUTINE FISHIN
(LARGE SCALE SYSTEMS STAT-PACK PROGRAMMERS REFERENCE SEC.13 PP.9-13).

METHOD :
****

JASPEN N. GIVES THE VALUE OFTHE VARIABLE Z WHICH IS NORMALLY DISTRIBUTED WITH ZERO MEAN AND UNIT VARIANCE, AS A FUNCTION OF THE VARIANCE RATIO F OF TWO INDEPENDENT VARIANCES WITH N1 AND N2 DEGREES OF FREEDOM.

```
          ((1-2/(9*N2))*F**(1/3) - (1-2/(9*N1)))
    Z=  ---------------------------------------
          ((2/(9*N2))*F**(2/3) + 2/(9*N1))**0.5
```

WE HAVE FOUND AN INVERSE TRANSFORMATION OF F AS A FUNCTION OF Z WHICH CONSISTS OF A SINGLE SOLUTION OF A SECOND DEGREE EQUATION IN F**(1/3) :

```
        AA*X*X + BB*X + CC = 0

WITH
                AA = C*Z*Z - A*A
                BB = 2*A*B
                CC = -B*B + D*Z*Z
                 X = F**(1/3)
AND

                A = 1 - 2/(9*N2)
                B = 1 - 2/(9*N1)
                C = 2/(9*N2)
                D = 2/(9*N1)

DELTA = BB*BB-4*AA*CC
IF DELTA > 0

X = (-BB-SQRT(DELTA))/(2*AA)

AND THE SOLUTION F = X*X*X
```

RESTRICTIONS :
****

1). N1 AND N2 GREATHER THAN OR EQUAL TO THREE
2). PROBABILITY LEVELS BETWEEN 0.1 AND 0.00001
    HAVE BEEN TESTED.

RESULTS OF THE TESTS :
****

WE HAVE PROCEEDED TO DIFFERENT TESTS FOR USUAL LEVELS OF PROBABILITY (FROM 0.1 TO 0.00001), THE DEGREES DEGREES OF FREEDOM BEING OBTAINED BETWEEN 3 TO 100 BY RANDOM GENERATORS. THE COMPARISON WAS DONE WITH THE SUBROUTINE FISHIN OF UNIVAC BY ESTIMATION OF THE QUANTITY

```
        D = ABS(F1-F2)/F2
WHERE   F1 IS F OF OUR METHOD (SUBROUTINE PROBI).
AND     F2 IS OBTAINED BY THE SUBROUTINE FISHIN.
```

THE MEAN AND STANDARD ERROR OF D WITH 100 TRIALS EACH ARE :

PROBABILITY	MEAN OF D	STANDARD ERROR
0.1	0.000074	0.000004
0.05	0.000134	0.000010
0.01	0.000202	0.000013
0.001	0.004739	0.003059
0.0001	0.062081	0.040453
0.00001	0.072889	0.044449

LISTING OF THE PROGRAMS : (SUBROUTINES PROBI AND NDTRI).
****

```
RTRAN IV      V01C-03A

           SUBROUTINE PROBI(F,P,N1,N2,IER)

 APPROXIMATION OF THE VALUE OF THE VARIANCE RATIO F ASSOCIATED WITH
 A PROBABILITY P AND N1 AND N2 DEGREES OF FREEDOM.

           REAL*8 A,B,C,D,AA,BB,CC,DELTA,X

 CALCULATION OF THE CRITICAL VALUE Z OF THE NORMAL DISTRIBUTION.

           CALL NDTRI(P,Z,D,IER)
     C
           IF(Z.LT.0)Z=-Z

 THIS APPROXIMATION IS VALID FOR N2 VALUES GREATHER THAN OR EQUAL
 TO THREE.

           IER=0
     C
           D=2./(9.*N1)
           C=2./(9.*N2)
           A=1-C
           B=1-D
     C
           AA=C*Z*Z-A*A
           BB=2*A*B
           CC=-B*B+D*Z*Z
     C
           DELTA=BB**2-4*AA*CC
     C
           IF(DELTA)1,2,3
     1     IER=1
           RETURN
     2     X=-BB/(2*AA)
           GO TO 4
     3     X=(-BB-DSQRT(DELTA))/(2*AA)
     4     IF(X.GT.0)GO TO 5
           IER=2
           RETURN
     5     F=X*X*X
           RETURN
           END
```

```
RTRAN IV      V01C-03A

           SUBROUTINE NDTRI(P,X,D,IER)

 DETERMINATION OF THE Z VALUE (X) AT A GIVEN LEVEL OF PROBABILITY P.

 REFERENCES : ABRAMOVITZ M. AND STEGUN L.A., HASTINGS C. (1955)

           A0=2.515517
           A1=0.802853
           A2=0.010328
           B1=1.432788
           B2=0.189269
           B3=0.001308
           PP=0.3989423
     C
           IER=0
           IF(P)1,4,2
     1     IER=-1
           GO TO 12
     2     IF(P-1.)7,6,1
     4     X=-1.E25
     5     D=0.
           GO TO 12
     6     X=+1.E25
           GO TO 5
     7     D=P
           IF(D-0.5)9,9,8
     8     D=1.-D
     9     T2=ALOG(1./(D*D))
           T=T2**0.5
           X=T-(A0+A1*T+A2*T2)/(1.+B1*T+B2*T2+B3*T*T2)
           IF(P-0.5)10,10,11
    10     X=-X
    11     D=PP*EXP(-X*X/2.)
    12     RETURN
           END
```

The following were developed by Alan Mazer and Ron Pipes. Questions about them can be answered by sending a SASE to:

Alan Mazer
U.S. Borax Research Corporation
412 Crescent Way
Anaheim, CA 92802

SQUEEZING WITHOUT DUP

There is now a way to do a type of device squeeze without really squeezing. The reason for developing this can be seen by looking at the way we use RT-11. We use the FB monitor to run a foreground job that reads data from three automated instruments, and copies the data from each to their associated TTY and associated data file on a floppy. This means the foreground has three files open at one time, and, since each instrument is at a different stage of analysis, stopping all three instruments to allow a squeeze would be time consuming. The solution is to copy files from the end of the disk to free space in the middle. Usually, moving a few files is enough to make a large enough free space. Back in RT-11 V2, PIP had an X option that would copy a file to itself, only at a different place on the disk. This was how a file could be moved to free up space. As PIP was "enhanced", the X option was discarded. We have brought back a way of moving files which is implemented in TECO.

The first part of the squeeze package is the file MOVE.TEC. This TECO macro take as an argument the name of the file to be moved. The macro reads the file into the buffer, sizes it, then allocates a file that is the correct size, and finally writes out to the file. The program uses the same file name, and does not make a backup. The means that a warning message (%Superseding existing file) is printed, but since there is no backup to delete, your space is already freed. This macro is limited because it only works on ASCII text files (no SAV or OBJ files) and the file must be small enough to fit in the buffer (this is so MOVE can determine its size). The macro will print an error message if the file is to big, or the file can not be found. You can use this macro to move one file at a time. It is used as follows:

Lets say you need 30 blocks, and your disk is fragmented. Here is part of a full directory listing :

```
              < UNUSED >      14
              WASTE .SPC       7
              < UNUSED >      20
```

If you save the monitor command:

```
              .ED/TEC/EXE:WASTE.SPC  MOVE
```

You would then have the following directory:

```
              WASTE .SPC       7
              < UNUSED >      34
```

You now have more than 30 blocks. As you can see, the macro is a useful tool for quick moves, but if you needed to do several, it might take a while. A way of automating it further is needed.

The next step is to do MOVEs on enough files to get the disk a little less fragmented. We found that, in general, the last part of the directory is the most fragmented. That means that a move of the last few files would clear up enough space, and this is, in our case, what happens. To automate this process, there is a TECO program, called SQUEEZ.TEC, that will read a directory listing from a file, pick up the number of files you want moved from the end of the directory, and generate a .COM file to execute the MOVEs on those files. When the program exits, it also starts the new command file. The number of files to be moved is determined by either by: 1) If the ED/EXE command has an argument (e.g. ED/EXE:10), that is used, or 2) If there is no argument, the program will ask the user how many files to move.

SQUEEZ changes a directory listing, in FILE.DIR, into the command file. It assumes that the listing has been made with the options /FAST and /COLUMN:1. To save time and thinking, a SQUEEZ.COM file was written that contains the directory command, the SQUEEZ macro start-up, and some other commands. We simply enter @SQUEEZ and its off and running. With these three files, we can do partial squeezes without interupting the foreground.

HOW THEY WORK (briefly)

```
MOVE:     Strip block size ( [n] ) from file name, if given.
          Put file name in F.
          Make file name into macro to read file, and put in O.
          Kill buffer. Execute O ( open file ). Read into buffer.
          Save pointer to end in 1. Append.
          Compare old pointer to new end, if different, file was to big.
          Determine file size.  Make write macro that allocates correct size.
          Execute write macro. Exit.

SQUEEZ:   Initialize counter C ( number of files to move ).
          If argument was passed, put in C, and tell # of files moved.
          If no argument passed, read from TTY and put in C.
          Open FILE.DIR for input and output ( so no .BAK is created ).
          Delete bottom two lines ( # files and # free blocks in listing ).
          Move up C lines. Kill from begining to pointer.
          Remove all blanks.
          Make each line into a command to ED/EXE:file MOVE.
          Insert commands to delete FILE.DIR, give a directory of free space,
          and set the TT to quiet.
          Exit and start FILE.DIR, the command file just created.
```

```
          .R TECO
          *ERMOVE.TEC$Y$HT$$
          !       EDIT 2 of  on 31-Jul-80 by VT52 TECO !
          !       EDIT 1 of  on 31-Jul-80 by VT52 TECO !
          !       EDIT 0 of MOVE.TEC[3] on 31-Jul-80 by VT52 TECO !
          ! CALL SEQUENCE:

          EDIT/EXE:file.ext MOVE

          !

          OJ:S[$"SRK' HXF OJI:ER$ZJ@I/$"F ^A
          ?MOVE-F-Output file not found - ^A:GF^A
          ^AHKEKEX'$$/ HXO HK MO Y ZU1 A ZJ Q1-."N ^A
          ?MOVE-F-File too big for move
          ^AHKEX'  Q1/512+1U1 OJI
          $OJGFI[$Q1\@I/]$$/ OJIEW$OJ 1XCOJ1KMC
          EX*^C$$


          .R TECO
          *ERSQUEEZ.TEC$Y$HT$$
          !       EDIT 12 of SQUEEZ.TEC on 27-Aug-80 by VT52 TECO !
          !       EDIT 0 of SQUEEZ.TEC[2] on 22-Aug-80 by VT52 TECO !
          OUC Z"NOJ\UC'
          QC"E @O!READIN! '
```

```
 ^ANumber of files to be moved = ^A QC=^A
 ^A @0!OPEN!
!READIN! ^ANumber of files to be moved ? ^A
 1#2#8#ETET < ^TUD QD-13"E 0;'
 QD-127"E QC/10UC 8^T 32^T8^T'
 QD"D QD^T (QC*10)+(QD-^^0)UC ' > ^TUD ^A
^A
!OPEN!   ERFILE.DIR$Y$EWFILE.DIR[2]$
ZJ-2K
ZJ-QCLB,.K
<:S $;FR$>
QC<IED/EXE:$LRRI MOVE$L>
IDELETE FILE.DIR/NOQUERY
DIR/FREE
SET TT QUIET
$EG@FILE.DIR$$*^C$$


.TY SQUEEZ.COM
!       EDIT 2 of SQUEEZ.COM on 27-Aug-80 by VT52 TECO !
!       EDIT 1 of  on 22-Aug-80 by VT52 TECO !
!       EDIT 0 of SQUEEZ.COM[2] on 22-Aug-80 by VT52 TECO !
SET TT NOQUIET
DEL *.BAK/L
DIR/FA/C:1/OUT:FILE.DIR
ED/EXE SQUEEZ
DIR/FR
        .@SQUEEZ

        .DEL *.BAK/L
        ?PIP-F-File not found DK:*.BAK

        .DIR/FA/C:1/OUT:FILE.DIR

        .ED/EXE SQUEEZ

        Number of files to be moved ? 3
        %Superseding existing file

        .ED/EXE:TS1187.JF2 MOVE

        %Superseding existing file

        .ED/EXE:TS1187.KFE MOVE

        %Superseding existing file

        .ED/EXE:SQUEEZ.DOC MOVE

        %Superseding existing file

        .DELETE FILE.DIR/NOQUERY

        .DIR/FREE
         16-Sep-80
        < UNUSED >      1              < UNUSED >      2
        < UNUSED >      2              < UNUSED >      1
        < UNUSED >      2              < UNUSED >      2
        < UNUSED >      1              < UNUSED >      1
```

```
        < UNUSED >      3              < UNUSED >      5
        < UNUSED >      3              < UNUSED >      2
        < UNUSED >      2              < UNUSED >      4
        < UNUSED >      5              < UNUSED >      3
        < UNUSED >      2              < UNUSED >      2
        < UNUSED >     10              < UNUSED >     16
        < UNUSED >      0              < UNUSED >      4
        < UNUSED >     16              < UNUSED >     26
         0 Files, 0 Blocks
         115 Free blocks

        .SET TT QUIET
```

Ian Hammond
HAMMOND-software
Am Feldborn 22
D-34 Goettingen
West Germany

DECUS Symposium, Amsterdam

The Amsterdam Symposium was well attended. The RT-11 training session, given by Anton Chernoff, overflowed its planned seating capacity. The session included tutorials on MTT and XM and some hours of questions and 'real' answers on V4.

The 'What to do when RT-11 runs out' session, planned to look at growth paths for to RT-11, turned into an RT-11/RSX-11 session since only the RT-11 and RSX-11 representatives attended. Some would have liked to hear more about RSTS - particulary since RSTS already has an RT-11 emulator. The official Digital reaction is that the 'RSTS RT-11 emulator is unsupported'. But then, some of Digitals best software is 'unsupported' (e.g. TECO). Indeed, the RT-11 development group use an 11/70 RSTS-E system to develop much of RT-11.

A later session handled RT-11/RSX-11 migration specifically. Ruediger Meisenburg (Heinrich Hertz Institute, Berlin) presented a very clear paper on moving a telephone network simulation from RT-11 to RSX-11. The main issue was mapping RT-11 SYSLIB calls into equivalent RSX-11 calls. This session concluded with a presentation of the RSX-11 RT-11 emulator. Les Parent (RT-11) and Stephen Paavola (RSX-11) stressed (repeatedly) that there was no committment to actually release the emulator. Digital polled potential support for the emulator - even about pricing. Educated opinion has it that the emulator passed its test of fire and probably will be released. The next version of RSX-11M should finish field-tests and be out in the third quarter of '81 (RSX does a year of field tests). This RSX-11M version will probably support DCLS console commands.

At the SIG business meeting Ray Carpenter (European RT-11 Sig Chairman) announced that the Mini-tasker was to become the official newsletter for the European area. Yours truly was appointed as European Correspondent to the Mini-tasker. European users can send their input to me to forward to Ken Demers. Hopefully, this wider distribution in Europe may mean that we will receive it faster - it usually takes 4 to 6 weeks.

DECUS Migration Working Group

A DECUS Working Group was formed at the symposium to examine the issues of migration, portability and compatibility. The group presently has RSX, RT and VAX representatives. The group plans to produce a roadmap of Digital operating systems, devices and options plus suggested techniques for improving program portability. Users who have ideas in this area should contact Ian Hammond, the group chairman, at the address above. We are looking for users who have had experience moving between systems.

Hidden RT-11

I have had some requests were for more information on 'hidden' RT-11 features. Here a summary of some features; perhaps other users can send additional information in this area to the Mini-tasker.

MAKE, MUNG, TECO - These three 'Standard TECO' commands are available in the RT-11/FB V4 monitor. Size constraints did not permit them to be supported in the RT-11/SJ monitor. However, a conditional change in KMON will extend support to SJ with a SYSGEN session (has anyone done this?).

PASCAL - The RT-11/FB monitor supports a PASCAL command. This command simply calls PASCAL with a copy of the command string - it is upto the PASCAL compiler to define and decode any parameters. One possible reason for a Digital RT-11 PASCAL command could be a Digital RT-11 PASCAL implementation?

RENAME - The RENAME command copies the device specification from the 'From' file to the 'To' file. That is:

```
 RENAME DX1:A.A B.B implies DX1:(A.A B.B) implies DX1:A.A DX1:B.B
```

This also applies to PIP /Rename operations. The Digital comment to this was 'Yes, we finally noticed that rename operations only take place on a single device'.

COPY/FILE/DEVICE - This feature is 'buried' rather than 'hidden' - it is often overlooked. COPY/FILES/DEVICE permits volume/file copy operations which are very useful for backups.

DOLLAR - The RT-11 Keyboard Monitor will ignore a dollar sign ($) in column zero of a console command or command file command. Its not worth reading to much into this, other than to note that it IS VAX/VMS compatible - i.e. VAX/VMS command procedures require a dollar in column zero. On the subject of command procedures, it appears that RT-11 Autopatch will contain a functional (but officially 'unsupported') version of RSX IND which handles conditionals and looping in command files. On the subject of Autopatch I guess we wont see it until RT-11·4.1 since it will probably require KMON modifications.

DIR/DUMP - RT-11 releases sometimes show evidence of work in progress. KMON has a entry for a DIR/DUMP command which is not implemented in DIR. V3 RT-11 had unused PRINT/DIR and TYPE/DIR command entries.

F4P - Since V3 there have been a couple of FORTRAN IV-plus commands in KMON. The SET FORTRAN F4P and FORTRAN/F4P commands would seem to indicate that F4P support was planned for RT-11. I guess it WAS but whether it still IS (after three years) is unknown. F4P needs an RSX-11-like RMS (Record Mangagement System) which RT-11 doesn't have. A lot of users

feel that F4P could be (and should be) the basic compatibility tool for moving programs around Digital systems - FORTRAN-IV-minus doesnt seem to fill that position. Questions to Digital on this subject are usually answered with "ask the FORTRAN development group".

Tricky RT-11

Another category of hidden RT-11 features are distinquished by the fact that they are potentially non-portable - sometimes they are plain dirty. Portability includes moving programs between different versions of RT-11 (V3, V4, V5 etc.), different configurations (SJ, FB, SYSGEN monitors) and to and from RT-11 emulators (RSTS, planned RSX, etc.). This raises the rather fuzzy issue of the 'RT-11 Architecture' (would the people in the back rows please stop laughing!).

The most clearly defined area of the RT-11 architecture is the program request set. The Keyboard Monitor command definitions are vague - but this is not so critical. Non-EMT monitor services are precarious. But, the most clearly dangerous area are the RT-11 monitor offsets and side-affects. Offsets tend to be change between releases, configurations and implementations. Further, if a feature is not 'defined' by the RT-11 documentation then the RT development group are free to change or remove it; Murphie's law tells us that they inevitably will.

RT-11 V3 and V4 have both included some mechanisms that clean up the architecture. V3 introduced .GVAL as a portable way to get the value of a monitor offset. V3 re-implemented .DATE as an EMT for the same reason. V4 introduced the .SDTTM EMT to set up the date and time - which means that a program is no longer dependent on the location or implementation of the psuedo-clock.

On the other side of the coin, RT-11 has become very dependent on job and monitor impure area offsets. V4 PIP, DIR, DUP and RESORC do a lot of bit-bashing in the monitor tables - more than V3. Now, if RT utilities need these services then its a safe bet that some user programs will also require them. But, the RT group are in a much safer position than users when it comes to betting on portability prospects.

There is a potential solution to this problem. A library of subroutines could be supplied that would handle the dirty operation on behalf of the program. Moving between implementations would just require linking to a local library. Obviously, it would be wonderful if the RT-11 group would provide such a library. If not, then perhaps we users should think about doing it.

Finally, here are two examples of RT-11 'dirty tricks":

KED - KED uses a very sneaky trick to get some scratch disk space. KED uses the final blocks of the tentative file allocated for output. Then, before it closes the file it fudges the 'highest block written' entry in the output files CSW area to ensure the closed file has the correct length. An ingenious trick - but no marks for portability. KED could have just as easily opened a temporary file.

USR COLLECT CALL - The RENAME/PROTECT and COPY operation that retains an file creation date are both implemented with a call back to a program from the USR with a register pointing at the directory entry for a file. The program (usually PIP) reads/modifies the directory entry's flag word (for PROTECT) or date word. This approach has the advantage that it doesnt cost much code in the already overcrowded USR. The USR is told that the collect call is required by setting an address in the request area odd and sending an extra argument that defines the address of the collect call subroutine. If you set that address odd by mistake you will get a crash that will be very difficult to debug.

RT-11/PDP-11 ASSEMBLY LANGUAGE TUTORIAL:
THE PHILOSOPHY AND STRUCTURE OF THE UNIBUS

Part 1 -- An Overview

by

Ray Strackbein
Cort Data Systems
77-920 Avenue of the States
Palm Desert, California 92260

THE ARCHITECTURE OF THE PDP/11

The PDP-11 family of mini-computers was originally designed to use only one bus -- the central processor unit (CPU), the memory, and all peripherals are directly attached to this bus. The primary reason for this was to allow all of the components of the computer system to 'talk' directly to each other -- not only can the CPU control its memory and all of its peripherals (as is universally the case in all computers I know of), but the all of the peripherals can 'talk' to the computer's memory and to each other as well. While it is not usually the case that one would want to have a magnetic disc talk directly to the console terminal, to do so is not prohibited by the design of the PDP-11.

The bus in a PDP-11 is called the Unibus. The Unibus transmits both address and data information (as well as other status and arbitration signals) along its length. To better understand the Unibus, let us look at just how the bus is used.

The Unibus is used to transmit information from one place in the computer system to another. Just as an example, let us take the case of memory transfering data to a disc. In order for this transfer to take place, three things must be known: 1) Where the data is coming from, 2) where the data is going to, and 3) what the data is. The data is usually a byte (8 bits) or a word (16 bits) of information and is carried along the 'DATA' lines of the Unibus in parallel form (one wire carries one bit of information -- there are 16 data wires in the Unibus: D0, D1, D2, ..., D13, D14, D15). Simple. To look at the data on the Unibus at any given time, we just look at the 16 Unibus Data wires.

Figuring out where the data is coming from and where it is going to is a bit more complex. In all data transactions there is always a bus master and a bus slave. Data can be transfered from the slave to the master, or from the master to the slave, so we must have two more wires: Data out (DATO) and Data in (DATI). If the data is out of the master the data out wire is asserted (DATO); if the data is out of the slave and into the master, the the Data in (DATI) wire is asserted. The secret to understanding this is to look at the master device. If the DATO is asserted, the the data in going Out of the master to the slave -- if the DATI (Data In) wire is asserted, then the data is going into the master from the slave.

We now get to try to figure out which device is the master and which device is the slave. The slave is easy ... its address is on the address lines of the Unibus. And the master? Well, its the device controlling the Unibus at the moment.

Now back to that example of data being transferred between disc and memory. Normally the central processor (CPU) is the bus master and memory is the slave. The CPU fetches an instruction from memory (the CPU is the master, the DATI line is asserted to show that the data is going from the slave to the master, and the memory address is on the Unibus address lines which tells what memory location to put its data on the Unibus). If that instruction happens to enable the disc controller, the controller waits until the CPU is between bus cycles and tries to assert the bus. The bus is so constructed that if several devices try to become bus master at the same time, the device of the highest priority will become bus master. In this case, our disc drive becomes the bus master. If this is a disc write, the address of the location in memory to send the data is put on the address lines and the DATI (and MSYN) is asserted. The slave (the asserted address in memory) responds by putting the contents of its memory location in the data lines and asserts SSYN which causes the disc controler to accept the data.

All of the devices on the Unibus must have its own address on the Unibus. There are 2 to the 15th power (32,768) words which may be addressed by the Unibus (by using memory management, four times this number (131,072)). To avoid contention between memory and peripheral devices for address space, DEC has specified that the top 4K words addressable by a computer system shall be reserved for use by peripherals. Memory may use the remaining address space.

The whole result of this scheme is that there is virtually no difference between memory and peripheral devices on the Unibus. The CPU can read and write peripheral registers just as it can read and write memory. This makes things very simple both for trouble-shooting and for designing peripherals.

DESIGNING PERIPHERALS

Let's say that we want to design a disc controller to interface to the Unibus. To transfer a block of data in memory through a disc we need to know: 1) The starting memory address of the transfer, 2) The number of words to be transferred, 3) the cylinder, track, sector and drive number for the start of the transfer, 4) possibly a word to buffer the data from or to the drive, and finally, 5) miscellaneous control, status, and error information.

It is a simple matter, then, to reserve five or six 16-bit addresses for our controller to use. One word might be used for the starting memory address, another to be used as a word counter, a third for use to pass the drive, cylinder, track, and sector number between the processor and the disc controller, a fourth as a data buffer, a fifth to pass control information (to tell the drive whether to read, write, format, or seek, and whether or not to interrupt the processor when a command has completed), and a sixth for the controller to pass error and status information back to the computer.

A look through DEC's Peripherals Handbook at some of the discs will show this idea in much more detail.

MEMORY MANAGEMENT

At this point, let's deviate slightly from the topic of the UNIBUS and try to explain just a little about memory management.

You should already know that at the heart of the operation of the PDP-11 computer is general register 7, also referred to as the program counter. This register always points to the address in memory of the next computer instruction (or part of an instruction) to be executed. Since the data path of the PDP-11 is only 16 bits wide, any register (including the program counter) can only point to a 16-bit address. If we are to try to address more than 16 bits worth of memory, some place has to be found to store some extra bits. That place is the memory management unit.

Because memory is most often addressed sequentially (especially in the case of memory which contains computer instructions) we can conclude that if the last memory reference was to location 'A', more than likely the next memory location to be addressed will be 'A+2'.

By partitioning all of the computer's memory into 32K-word partitions, we can tell the memory management unit what partition we are in and then let all references to memory refer to a relative location within that partition. Because of the large size of the 32K-word partition and because of the fact that most often memory is addressed sequentially, we will have to spend very little time juggling memory-management pointers because we will usually be referring to the same partition we were referring to in the previous cycle.

This whole idea tends to fall apart whenever an interrupt occurs. There is no correlation whatsoever between the address (or partition) we were accessing before the interrupt occured and the address we will access as a result of that interrupt. But the solution is very easy.

You should know that an interrupt causes the contents of R7 (the program counter) to be pushed on the stack. In addition, the Processor Status Word is also put on the stack by an interrupt. The PSW and R7 are then loaded with the contents of the two-word vector pointed to by that interrupt.

Because the PSW had a few extra bits which weren't being used, someone thought of using a few of those extra bits to control the memory management unit. The result is that whenever an interrupt is taken, the memory management unit is immediately notified by the bits being changed in the processor status word. This allows the processor to rapidly cross partition bounds if necessary whenever an interrupt occurrs.

Because a trap instruction (BPT, IOT, EMT, and TRAP) is nothing more than an interrupt generated by the CPU, the RT-11 operating system and its layered products use the various trap instructions for calling system routines -- the memory management unit is adjusted by its hardware to account for the occurance of an interrupt and no extensive software adjustments are necessary.

Because a DMA data transfer from a peripheral would be severely slowed if the transfer had to tell the memory management unit it wanted to stuff data in a partition different from the partition currently being used by the CPU, a peripheral can directly address 128K-words of memory through the Unibus through the memory extention bits. A DMA transfer does not use the memory management unit.

RT-11/PDP-11 ASSEMBLY LANGUAGE TUTORIAL
FALL DECUS, SAN DIEGO, NOVEMBER 7, 1980
BY
RAY STRACKBEIN
CORT DATA SYSTEMS
77-920 AVENUE OF THE STATES
PALM DESERT, CA. 92260

```
                ;THESE ARE TEST PROGRAMS FOR USE IN TESTING AND DEBUGGING
                ;YOUR SYSTEM
                ;
                ;
                ;THIS PROGRAM WILL CAUSE A TERMINAL OR OTHER SERIAL DEVICE
                ;TO ECHO
                ;
        105737  ECHO:   TSTB @#177560
        177560
        100375          BPL ECHO
        105737  1$:     TSTB @#177564
        177564
        100375          BPL 1$
        113737          MOVB @#177562,@#177566
        177562
        177566
        000766          BR ECHO
                ;
                ;THIS PROGRAM WILL TEST MEMORY
                ;
        010702  MEMTST: MOV PC,R2
        005000  1$:     CLR R0
        011001  2$:     MOV (R0),R1
        005110          COM (R0)
        005101          COM R1
        021001          CMP (R0),R1
        001007          BNE OOP
        005110          COM (R0)
        005101          COM R1
        022001          CMP (R0)+,R1
        001003          BNE OOP
        020002          CMP R0,R2
        103765          BLO 2$
        000763          BR 1$
        000000  OOP:    HALT
                ;
                ;IF NOTHING RUNS, SOME PROCESSORS ALLOW INSTRUCTIONS TO BE
                ;STORED IN THE COMPUTER'S GENERAL REGISTERS.  ALL MEMORY AND
                ;PERIPHERALS CAN BE REMOVED FROM THE COMPUTER AND THE COMPUTER
                ;CAN STILL RUN A TRIVIAL PROGRAM SUCH AS:
                ; .=177700      ; THIS IS THE ADDRESS OF R0
        000240          NOP
        000777  1$:     BR 1$
                ; IN THIS CASE A 240 (THE NO-OP INSTRUCTION IS PLACED IN R0,
                ; AND A 777 (NOTE NOT A 776) IS PLACED IN R1. IF THE ADDRESS
                ; OF R0 (177700) IS LOADED INTO THE SWITCH REGISTER AND
                ; STARTED, THE COMPUTER WILL ACTUALLY RUN WITHOUT MEMORY.
                ;
                ;
                ; ANOTHER USEFUL TOGGLE PROGRAM USING ONLY THE GENERAL
                ; REGISTERS IS A PROGRAM TO LOAD EACH MEMORY LOCATION WITH
                ; ITS OWN ADDRESS:
                ; .=177700
        005005  START:  CLR R5
        000240          NOP
        010525          MOV R5,(R5)+
        000777  1$:     BR 1$
                ;
                ; IF ONE FEELS THAT THE COMPUTER'S MAIN MEMORY MIGHT BE A
                ; BIT UNSTABLE, ONE CAN LOAD MEMORY WITH ITS OWN ADDRESS AND
                ; LOOP THROUGH A READ AND COMPARE PROGRAM STORED IN MAIN
                ; MEMORY SUCH AS:
        010705  MTEST2: MOV PC,R5
        005000          CLR R0
        010020  1$:     MOV R0,(R0)+
        020005          CMP R0,R5
        103775          BLO 1$
        005000  2$:     CLR R0
        020020  3$:     CMP R0,(R0)+
        001003          BNE OOPS
        020005          CMP R0,R5
        103774          BLO 3$
        000772          BR 2$
        000000  OOPS:   HALT
```

DEC INPUT

RT-11 TRAINING PROGRAM (Continued)

CHICAGO (312) 640-5520

Canadian Customers: All registrations for U.S. courses should be made through the Ottawa (Kanata) Registrar, (613) 592-5111.

	Course #	Title	Length	Start Day	Tuition	Training Credit	October				November				December					January				February				March				
							6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30
P	EY-J2114-A0	Introduction to Minicomputers A/V	~5		$ 400	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2016-A0	Introduction to Minicomputers	5	MO	520	1				●			●				●			●				●						●		
P	EY-J2116-A0	Introduction to the PDP-11 A/V	~5		450	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2024-A0	PDP-11 Assembly Language Programming	5	MO	580	1	●		●		●	●			●					●			●			●			●			●
P	EY-J2050-A0	Programming in MACRO-11	5	MO	630	1		●		●		●	●			●				●				●	●		●			●		●
P	EY-J2048-A0	PDP-11 Concepts	3		460	1																										
	EY-J2000-A0	RT-11 User	5	MO	630	1						●																				
	EY-J0016-A0	Programming in FORTRAN IV	5	MO	630	1			●							●						●							●			
	EY-J2002-A0	RT-11 Programmer	5	MO	630	1							●																			
	EY-J2004-A0	RT-11 System Programmer	3		520	1																										

RT-11 CUSTOMER TRAINING PROGRAM

SAN FRANCISCO (408) 727-0200 Ext. 2142

	Course #	Title	Length	Start Day	Tuition	Training Credit	October				November				December					January				February				March				
							6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30
P	EY-J2114-A0	Introduction to Minicomputers A/V	~5		$ 400	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2016-A0	Introduction to Minicomputers	5	MO	520	1				●					●					●			●				●					
P	EY-J2116-A0	Introduction to the PDP-11 A/V	~5		450	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2024-A0	PDP-11 Assembly Language Programming	5	MO	580	1		●			●		●			●					●			●				●			●	
P	EY-J2050-A0	Programming in MACRO-11	5	MO	630	1			●			●					●					●			●				●			
P	EY-J2048-A0	PDP-11 Concepts	3		460	1																										
	EY-J2000-A0	RT-11 User	5	MO	630	1	●								●					●										●		
	EY-J0016-A0	Programming in FORTRAN IV	5	MO	630	1					●													●								
	EY-J2002-A0	RT-11 Programmer	5	MO	630	1	●								●					●										●		
	EY-J2004-A0	RT-11 System Programmer	3	MO	520	1					●													●								

WASHINGTON, D.C. (301) 459-7900 Ext. 2580, 2582

							6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30
P	EY-J2114-A0	Introduction to Minicomputers A/V	~5		$ 400	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2016-A0	Introduction to Minicomputers	5	MO	520	1				●						●						●					●					
P	EY-J2116-A0	Introduction to the PDP-11 A/V	~5		450	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2024-A0	PDP-11 Assembly Language Programming	5	MO	580	1	●		●		●		●		●		●			●		●		●		●		●		●		
P	EY-J2050-A0	Programming in MACRO-11	5	MO	630	1		●		●		●				●					●		●		●		●				●	
P	EY-J2048-A0	PDP-11 Concepts	3	MO	460	1							●										●						●			
	EY-J2000-A0	RT-11 User	5	MO	630	1	●													●												
	EY-J0016-A0	Programming in FORTRAN IV	5	MO	630	1									●						●							●				
	EY-J2002-A0	RT-11 Programmer	5	MO	630	1		●																								
	EY-J2004-A0	RT-11 System Programmer	3		520	1																										

NEW YORK (212) 971-3545

							6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30
P	EY-J2114-A0	Introduction to Minicomputers A/V	~5		$ 400	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2016-A0	Introduction to Minicomputers	5	MO	520	1			●						●					●							●				●	
P	EY-J2116-A0	Introduction to the PDP-11 A/V	~5		450	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2024-A0	PDP-11 Assembly Language Programming	5	MO	580	1			●						●					●			●				●					●
P	EY-J2050-A0	Programming in MACRO-11	5	MO	630	1	●			●						●					●			●				●				
P	EY-J2048-A0	PDP-11 Concepts	3		460	1																										
	EY-J2000-A0	RT-11 User	5	MO	630	1	●																									
	EY-J0016-A0	Programming in FORTRAN IV	5	MO	630	1											●							●					●			
	EY-J2002-A0	RT-11 Programmer	5	MO	630	1																										
	EY-J2004-A0	RT-11 System Programmer	3		520	1																										

BOSTON (617) 275-5000 Ext. 2380

							6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30
P	EY-J2114-A0	Introduction to Minicomputers A/V	≈5		$ 400	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2016-A0	Introduction to Minicomputers	5	MO	520	1	●			●			●		●		●			●		●		●				●		●		
P	EY-J2116-A0	Introduction to the PDP-11 A/V	≈5		450	1									CON	TIN	UOU	SLY	AVA	LA	BLE	AT	FACI	TY								
P	EY-J2024-A0	PDP-11 Assembly Language Programming	5	MO	580	1	●		●		●		●				●				●		●		●		●		●		●	
P	EY-J2050-A0	Programming in MACRO-11	5	MO	630	1	●	●			●	●	●		●	●				●		●		●		●		●		●		●
P	EY-J2048-A0	PDP-11 Concepts	3	MO	460	1							●				●						●							●		
	EY-J2000-A0	RT-11 User	5	MO	630	1		●			●					●				●							●		●			
	EY-J0016-A0	Programming in FORTRAN IV	5	MO	630	1			●						●								●						●			
	EY-J2002-A0	RT-11 Programmer	5	MO	630	1			●								●				●									●		
	EY-J2004-A0	RT-11 System Programmer	3	WE	520	1				●												●										

USER REQUESTS

MAX-PLANCK-INSTITUT
FÜR BIOCHEMIE
Ludwig Voet

8033 MARTINSRIED BEI MÜNCHEN
TELEFON (089) 8585-1
DURCHWAHL 8585-..........
TELEX 521740 mplb d

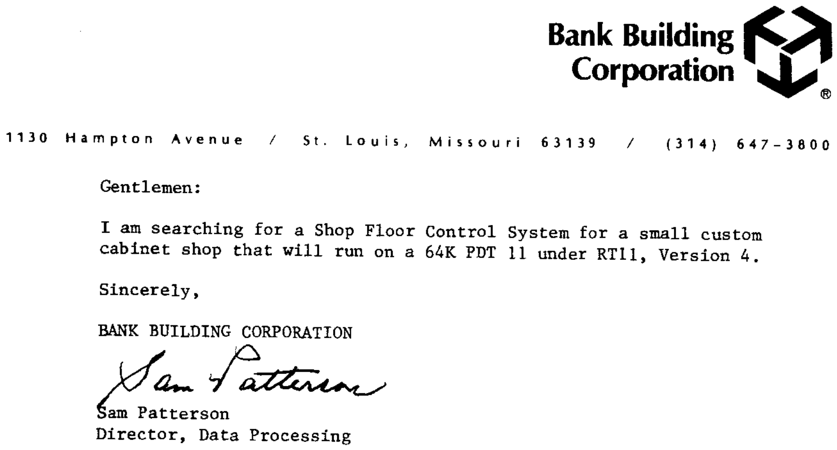
den 24. Okt. 1980

Dear Mr. Demers,

we are collecting data under RT-11 on Mag-Tape with the MT-Handler. Now we try to read these data on VAX-11/780, but there is no direct way to do this in RT-11 or VMS Operating System via Mag-Tape. Only RX-01 Floppy-Disc is supported by FILEX under RT-11 and VMS via DOS. Due to the large data set, this is a very uncomfortable way to do this.
We hope you or one of your readers know about existing written software to handle this problem.

Sincerely
L. Voet

Ludwig Voet
Max-Planck-Institut für Biochemie
D-8033 Martinsried bei München
West-Germany



Bank Building Corporation

1130 Hampton Avenue / St. Louis, Missouri 63139 / (314) 647-3800

Gentlemen:

I am searching for a Shop Floor Control System for a small custom cabinet shop that will run on a 64K PDT 11 under RT11, Version 4.

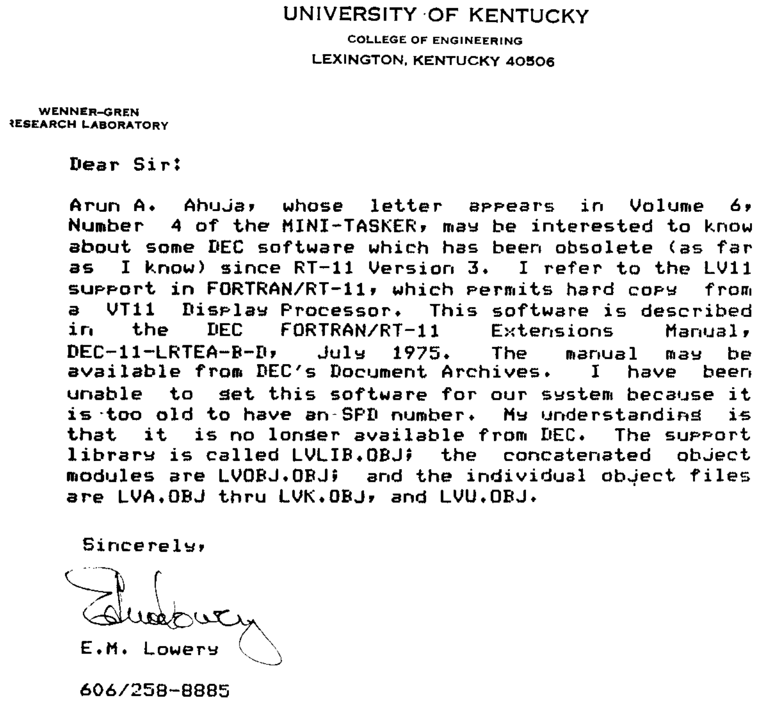
Sincerely,

BANK BUILDING CORPORATION

Sam Patterson

Sam Patterson
Director, Data Processing

USER RESPONSES



UNIVERSITY OF KENTUCKY
COLLEGE OF ENGINEERING
LEXINGTON, KENTUCKY 40506

WENNER-GREN
RESEARCH LABORATORY

Dear Sir:

Arun A. Ahuja, whose letter appears in Volume 6, Number 4 of the MINI-TASKER, may be interested to know about some DEC software which has been obsolete (as far as I know) since RT-11 Version 3. I refer to the LV11 support in FORTRAN/RT-11, which permits hard copy from a VT11 Display Processor. This software is described in the DEC FORTRAN/RT-11 Extensions Manual, DEC-11-LRTEA-B-D, July 1975. The manual may be available from DEC's Document Archives. I have been unable to get this software for our system because it is too old to have an SPD number. My understanding is that it is no longer available from DEC. The support library is called LVLIB.OBJ; the concatenated object modules are LVOBJ.OBJ; and the individual object files are LVA.OBJ thru LVK.OBJ, and LVU.OBJ.

Sincerely,

E.M. Lowery

606/258-8885

****

Spring 81 DECUS/US Symposium
RT-11 SIG Tape

HELP WANTED

For the next symposium in Maimi I need some help for building the master RT-11 tape. I am looking for a local shop that would be willing to permit me to use their equipment for building our tape. I'll need a minimum system having RX01's or RX02's and 9-track mag tape (800 bpi, 2400 ft spools). A larger disk such as an RK05, RL01 or RL02 would be very desirable. If necessary I can bring my own bootable RK05 disk cartridge. Actual use of the system would be during the evening hours. If anyone is willing to volunteer their shop and services please contact me.

Nick Bourgeois / 1738
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-8088

****

PAST SYMPOSIUM INFORMATION

****

Fall 80 DECUS/US Symposium
RT-11 Can Do It Workshop

This time we handled the workshop session in a more lively fashion than in the past. The workshop was run by three co-chairmen: Greg "SET SY CRASH" Adams, Ray "Gong" Strackbein and Nick "Reverand" Bourgeois. User input kept this fun forum alive and kicking for about five hours. Most of the session was taped. We'll try to have a summary for the next issue of the "Mini-Tasker".

****

STATUS OF RUNOFF (M02.2) FOR RT-11

This is an update on the RUNOFF M02.1 for RT-11 which I submitted to the Spring 1980 RT-11 SIG tape. I wrote in to Ken Demers shortly thereafter to follow up on that submission, and the letter was published in the June, 1980 Mini-Tasker.

First and foremost, this is a complete distribution of the latest DECUS library RUNOFF, version M02 (DECUS 11-241) as re-released and announced in DECUScope March 1980. Many, many bugs have been fixed in this version of RUNOFF, from the old M01 version. It represents Mark Lewis' (the 'M' in 'M02') last shot at it before he left for IBM land.

Besides the files needed to support RSX-11M/M+, IAS, and RSTS/E, I have added those needed to support RT-11. I have made no functional changes to RUNOFF, nor do I feel I should, as this makes the first time everybody in PDP/VAX land has had a common version of RUNOFF. I plan to submit this to the DECUS library shortly, with the fixes outlined below, as M02.2. It will also be on the Fall '80 RT-11 SIG tape.

After putting the M02.1 kit on the Spring '80 RT tape, I received a number of inquiries and a few bug reports. The main bug was that under certain conditions, RUNOFF ate up all of the stack and died. This was an omission on my part in RNRT11.MAC, where I failed to initialize the variable 'NLPG'. Also reported was the malfunctioning of "HEADER LEVEL" when "HEADERLEVEL" and "HL" worked. This was fixed in the command tables, where there were some entries out of sequence. Finally, the leading and trailing formfeeds were eliminated, as this is not normal practice on RT-11, where the LP:/LS: handlers usually are set to issue their own.

There is (rumored) a DEC Standard RUNOFF coming. I don't know when they're going to announce it, but it is at least running in prototype form on a VAX at Tewksbury. I don't know how different it will be from this version.

Finally, my thanks to Chester and Lois Wilson for their nice letter and offering of the 'Australian RUNOFF', an M01 RT-11 conversion. They noted several bugs, most of which were among those fixed in the M02 release. They also outlined some very good suggestions for enhancements to RUNOFF. I will present these at the next meeting of the text processing SIG.

If anyone wishes to contact me, they may do so at:

Robert B. Denny, dba
Creative System Design
3452 E. Foothill Blvd. Suite 601
Pasadena, CA 91107
(213) 792-9474

RT-11 V4 WITH NO DEC COMPATIBLE DISKS
By Bob Denny

I have a client who has a system with no DEC compatible disk units, but he does have a TM-11 emulating magtape unit. We got his first RT-11 system going a few years back by writing a driver for his nonstandard disk system, modifying the BSTRAP.MAC component of the RT-11 V3B monitor to include the bootstrap driver for his disk, physically taking his disk system to a site that had RK-05's, plugging it into the UNIBUS and getting a bootable system going. He takes his BPUS on magtape, and there has been no problem re-syggening since then.

Version 4 introduced some good news and some bad news. The good news is that there is no longer any device dependent code in the monitor. No more mods to BSTRAP.MAC! The bad news is that there is no way to get the distributed V4 system running unless you have a DEC compatible disk system, regardless of the distribution medium, even if you have the magtape distribution kit. The V4 DUP program won't run under earlier versions, even if you fake the version number in the monitor's offset area (it starts up but acts wierd). So there's no way to do a BOOT or COPY/BOOT of the new system, with a newly converted non-standard handler.

Not wanting to repeat the hassle of hauling his disk to the other site, we looked for a way to get going. We knew that we had to change the handler to use the deluxe macros furnished with V4 (thanks Anton!), and add the bootstrap support code. That was not too tough, and most of the bootstrap code was the same as that which we edited into V3B's BSTRAP.MAC.

Taking our best shot at that (we couldn't test it), we dove into the Software Support Manual (REALLY thanks!!!) to find out how the new V4 device independent boot system worked. It took several passes through section 7.10 to get it crystal clear. Now all we had to do was fake a "BOOT ddn:", "BOOT ddn:filnam", or a "COPY/BOOT xxn:filnam ddn:" process once, and we had our running V4 distributed monitor (assuming the handler worked).

We decided that faking the COPY/BOOT would be the easiest to check out, since we could inspect the disk in blocks 0-5 after running our 'faker' routine and verify that things looked per Fig. 7-8 in the SSM. Any of the BOOT processes would just die with few clues as to the cause, if there was a problem anywhere.

Following this writeup is a listing of the V4CPYB.MAC program which was written for this purpose. We copied the basic V4 distributed monitor components and handlers, including our new system device handler, to drive 0 of the nonstandard disk, while running V3B on drive 1. Then we ran V4CPYB. Next we checked the disk on drive 0 with DUMP to see that it looked like it should per the SSM, sec. 7.10.3.2. It did. We then hardware booted drive zero, and ... it died. After fixing a dumb error in the V4 converted driver, we replaced it on drive zero, reran V4CPYB, and hardware booted drive 0 again. This time we were greeted by the distributed SJ monitor! What fun!

If you want to know more, or have a question, feel free to write or call me. This program will be on the Fall 1980 RT-11 SIG tape.

Robert B. Denny, dba RBD
Creative System Design
3452 E. Foothill Blvd. Suite 601
Pasadena, CA 91107
(213) 792-9474

```
        .TITLE  V4CPYB
        .IDENT  "Y1.0"
        .ENABL  LC
;
; This program does a COPY/BOOT, Version 4 style, while you are
; running under an earlier version, such as V3B. It is used only to
; get a first V4 system booted on a system where there are no DEC
; compatible random access mass storage devices. I used it to get
; V4 going on a system which had only compatible 9 track tape.
; NOTE:
;       There are several places that must be edited prior to
;       running this program. They are marked with ;*** style
;       comments and explanation.
;
        .mcall  .lookup .enter  .readw  .writw  .exit
        .mcall  .print  .ttyin  .close  .hreset .fetch
;
; Absolute section for boot code.
;
        .asect  ; <<< reserve space for secondary boot >>>
        . = 1000
secbot:                         ; Secondary boot goes here.
        .blkw   <5*400>         ; First 5 blocks of monitor
pribot:
        .blkw   <3*400>         ; Space for primary boot
;
; ENTRY POINT.
;
        .psect  .prog.          ; Start the relocatable code
START::
        .print  #stmss          ; Ask user if he's sure
        .ttyin                  ; Get Y/N
        bic     #177640,r0      ; Make sure it's upper case
        cmp     r0,#'N          ; If 'N'
        beq     5$              ;   exit
        .fetch  #hspace,#dskblk ; Fetch the handler
        bcc     2$              ; If error
        .print  #dnfmss         ;   inform user
        .exit                   ;   and exit
2$:     .lookup #dpb,#0,#monblk ; Else, open the monitor file
        bcc     10$             ; If error
        .print  #mnfmss         ;   inform user
5$:     .exit                   ;   and exit
10$:                            ; Else, read blocks 0-4 of monitor
```

```
          .readw  #dpb,#0,#secbot,#<5*400>,#0
          bcc     20$             ; If error
          .print  #mremss         ;   inform user
          .exit                   ;   and exit
  20$:                            ; Else, close monitor
          .close  #0
          .lookup #dpb,#0,#devblk ; Lookup the handler file
          bcc     30$             ; If error
          .print  #hnfmss         ;   inform user

        .exit                   ;   and exit
30$:                            ; Else read primary boot
        mov     #pbaddr,r1      ; Copy primary boot address
        asr     r1              ; convert to word offset in file
        bic     #377,r1         ; and thence to block # where pri boot starts
        swab    r1
        .readw  #dpb,#0,#pribot,#<2*400>,r1 ; Read enough to get primary boot
        bcc     40$             ; If error
        .print  #hremss         ;   inform user
        .exit                   ;   and exit
40$:                            ; Else, move pri. boot into beg of sec. boot
        .close  #0              ; Close the handler file
        mov     #pbaddr,r1      ; Convert primary boot address to offset in
        bic     #177000,r1      ;   the 'pribot' buffer, then to the absolute
        add     #pribot,r1      ;   address of the start of the primary boot.
        mov     #secbot,r0      ; R0 --> start of secondary boot image
        mov     #pbsize/2,r2    ; R2 = #words in primary boot
50$:    mov     (r1)+,(r0)+     ; Copy primary boot to secondary boot image
        dec     r2
        bne     50$
        mov     monblk+2,@#secbot+4724 ; Set up the filename for the monitor
        mov     monblk+4,@#secbot+4726 ; in it's special spot.
;
; Now write out the boot data to the disk.
;
        .lookup #dpb,#0,#dskblk ; Non-filestructured open on the disk
        .writw  #dpb,#0,#secbot,#400,#0 ; Primary boot to block 0
        .writw  #dpb,#0,#secbot+1000,#<4*400>,#2 ; Sec. boot -> blks 2-5
        .close  #0
        .exit
;
; Program's data base. Edit ;*** sections for your non standard device
;  i.e., replace all 'xxn' and 'ddn' with 'SM0' and 'dd' with 'SM' to
;  COPY/BOOT on non-standard device SM0:
;
        pbaddr = ????           ;*** ddBOOT address from handler load map
        pbsize = 1000           ;*** Size of p-section ddBOOT from the
                                ;***  MACRO-11 listing of the handler.
        .nlist  bex
monblk: .rad50  "xxnRT11SJSYS"  ;*** xxn:RT11SJ.SYS is monitor to be booted
devblk: .rad50  "ddndd    SYS"  ;*** ddn:dd.SYS is to be system device
dskblk: .rad50  "ddn         "  ;*** ddn: where boot stuff goes.
dpb:    .blkw   6               ; DPB area for programmed requests
stmss:  .ascii  "COPY/BOOT xxn:RT11SJ.SYS ddm: [Y/N]?"<200> ;*** EDIT THIS TOO
dnfmss: .asciz  "V4CPYB - Device handler missing"
mnfmss: .asciz  "V4CPYB - Monitor file not found"
mremss: .asciz  "V4CPYB - Error reading monitor file"
hnfmss: .asciz  "V4CPYB -.Handler file not found"
```

```
hremsg: .asciz  'V4CPYB - Error reading handler file'
        .list   bex
        .even
hspace:                         ; Handler goes here.
        .end    start
```

****

DECUS Fall 80
RT-11 SIG Tape Submissions

****

N. A. Bourgeois, Jr.
Sandia National Laboratories
Systems Engineering Division 1738
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-8088

Patches through Aug 80 as published in the "Software Dispatch" for RT-11 V4.0 and BASIC-11 V2. See the file PATCH.DIR for more information.

Short paper describing an RT-11 application. Presented at the "How to Solve the Operating System Labyrinth" session. Files named RTAPPL.(RNO,DOC).

Short paper on RT-11 interrupt service routines by the rules and otherwise or moral and immoral. Presented at the "RT-11 Can Do It Workshop". Files named RTISRS.(RNO,DOC).

Short paper describing a library implementation of FORTRAN callable programmed requests for TSX/TSX-Plus. Presented at the "RT-11 Can Do It Workshop". See TSXLIB.DIR for more information.

MACRO-11 program to rewind and take off line all mag tape drives on a TM11 controller. File named RWMT.MAC.

MACRO-11 program to force the user to enter a valid date and time. File named SIGNON.MAC.

MACRO-11 program to prevent user from interfering with planned sequence of operations. File named WATCH.MAC.

```
RTAPPL.RNO    14  15-Oct-80      RTAPPL.DOC    18  16-Oct-80
RTISRS.RNO    17  15-Oct-80      RTISRS.DOC    23  16-Oct-80
RWMT  .MAC     8  07-Jul-80      SIGNON.MAC    41  07-Jul-80
WATCH .MAC    12  07-Jul-80      README.RNO     4  24-Oct-80
README.NAB     4  24-Oct-80
 9 Files, 141 Blocks
```

```
TSXLIB.DIR     1  16-Oct-80      TSXLIB.RNO    26  13-Oct-80
TSXLIB.DOC    36  16-Oct-80      TSXLIB.COM     2  08-Oct-80
MSGCOM.MAC     9  08-Oct-80      RCDLOK.MAC    10  08-Oct-80
TSXMSC.MAC     8  10-Oct-80
 7 Files, 92 Blocks
PATCH .DIR     8  16-Oct-80      RTPAT .RNO     4  06-Aug-80
BASPAT.RNO     4  26-Aug-80      RTPAT .DOC     4  06-Aug-80
BASPAT.DOC     5  26-Aug-80      DIRPAT.COM     1  04-Aug-80
DUP   .001     1  05-Aug-80      DIR   .001     1  05-Aug-80
LINK  .001     1  05-Aug-80      LIBR  .001     1  05-Aug-80
SIPP  .001     1  05-Aug-80      RT11SJ.001     1  06-Aug-80
070201.COM     1  05-Aug-80      070301.COM     1  05-Aug-80
070901.COM     1  05-Aug-80      071001.COM     1  05-Aug-80
071601.COM     1  05-Aug-80      010101.COM     1  05-Aug-80
RMONFB.001     1  05-Aug-80      BSTRAP.001     1  05-Aug-80
RMONFB.002     1  05-Aug-80      BSTRAP.002     1  05-Aug-80
010103.COM     2  05-Aug-80      BSTRAP.003     1  05-Aug-80
KMON  .001     1  05-Aug-80      RMONSJ.001     1  05-Aug-80
RMONFB.003     1  05-Aug-80      MTTINT.001     1  05-Aug-80
010104.COM     1  06-Aug-80      USR   .001     1  05-Aug-80
BSTRAP.004     1  05-Aug-80      RT11BL.001     1  05-Aug-80
RT11FB.001     1  05-Aug-80      RT11FB.002     1  05-Aug-80
BSTRAP.005     1  05-Aug-80      DD    .001     1  05-Aug-80
DM    .001     1  05-Aug-80      TM    .001     1  05-Aug-80
TJ    .001     1  05-Aug-80      TS    .001     1  05-Aug-80
010105.COM     2  06-Aug-80      060401.COM     1  06-Aug-80
060601.COM     1  06-Aug-80      010102.COM     2  06-Aug-80
PATA1 .MAC     1  25-Aug-80      PATA2 .MAC     1  25-Aug-80
350102.COM     1  25-Aug-80      350103.COM     1  25-Aug-80
PATB1 .MAC     1  25-Aug-80      PATB2 .MAC     1  25-Aug-80
350104.COM     1  25-Aug-80      PATC1 .MAC     1  25-Aug-80
PATC2 .MAC     1  25-Aug-80      350105.COM     1  25-Aug-80
PATD1 .MAC     1  25-Aug-80      PATD2 .MAC     1  25-Aug-80
350106.COM     1  25-Aug-80      PATE1 .MAC     1  25-Aug-80
PATE2 .MAC     1  25-Aug-80      350107.COM     1  25-Aug-80
PATF1 .MAC     1  25-Aug-80      PATF2 .MAC     1  25-Aug-80
350108.COM     1  25-Aug-80      PATG1 .MAC     1  25-Aug-80
PATG2 .MAC     1  25-Aug-80      350109.COM     1  25-Aug-80
PATH1 .MAC     1  25-Aug-80      PATH2 .MAC     1  25-Aug-80
PATI1 .MAC     1  25-Aug-80      350111.COM     2  25-Aug-80
PATJ1 .MAC     1  25-Aug-80      PATJ2 .MAC     1  25-Aug-80
PATJ3 .MAC     1  25-Aug-80      PATJ4 .MAC     1  25-Aug-80
350113.COM     1  25-Aug-80      PATL1 .MAC     1  25-Aug-80
PATL2 .MAC     1  25-Aug-80      350114.COM     1  25-Aug-80
PATM1 .MAC     1  25-Aug-80      PATM2 .MAC     1  25-Aug-80
350115.COM     1  25-Aug-80      PATN1 .MAC     1  25-Aug-80
PATN2 .MAC     1  25-Aug-80      350117.COM     1  25-Aug-80
PATO1 .MAC     1  25-Aug-80      PATO2 .MAC     1  25-Aug-80
350110.COM     1  25-Aug-80      PATI2 .MAC     1  25-Aug-80
350202.COM     1  27-Aug-80      SUCVT1.001     1  27-Aug-80
BASPAT.COM     2  27-Aug-80      SUCVT1.002     1  27-Aug-80
350201.COM     1  27-Aug-80      DUP   .002     1  21-Aug-80
FILEX .001     1  21-Aug-80      KED   .001     2  02-Sep-80
KED   .002     3  26-Aug-80      KED   .003     2  03-Sep-80
K52   .001     2  22-Aug-80      K52   .002     3  22-Aug-80
K52   .003     2  22-Aug-80      LINK  .002     1  21-Aug-80
LP    .001     1  28-Aug-80      LS    .001     1  28-Aug-80
RESORC.001     1  21-Aug-80      SRCCOM.001     1  21-Aug-80
TS    .002     1  28-Aug-80      061201.COM     1  05-Sep-80
```

```
061301.COM     1  05-Sep-80     062002.COM     2  05-Sep-80
070202.COM     1  05-Sep-80     070501.COM     1  05-Sep-80
070902.COM     1  05-Sep-80     071101.COM     1  05-Sep-80
071201.COM     1  05-Sep-80     170101.COM     1  05-Sep-80
170102.COM     1  05-Sep-80     OCHAIN.MAC     1  05-Sep-80
ICHAIN.KED     2  05-Sep-80     170103.COM     1  05-Sep-80
170201.COM     1  05-Sep-80     170201.TST     1  05-Sep-80
170202.COM     1  05-Sep-80     ICHAIN.K52     2  05-Sep-80
170203.COM     1  05-Sep-80     062001.COM     2  05-Sep-80
CONHOG.MAC     2  05-Sep-80     170101.TST     1  05-Sep-80
RTPAT .COM     3  05-Sep-80     170102.TST     2  05-Sep-80
170202.TST     2  05-Sep-80
 131 Files, 173 Blocks
```

****

Michael N. Levine
Naval Weapons Ctr
Code 3513
China Lake, CA 93555
(714) 939-2417

INDEX IS A FORTRAN CROSS REFERENCING PROGRAM. A FORTRAN SOURCE FILE PROCESSED BY INDEX WILL BE CHECKED FOR ALL OF ITS VARIABLE NAME AND LABEL USEAGE. THE RESULTS WILL THEN BE LISTED IN ALPHABETICAL ORDER, LISTING ALL THE VARIABLE NAMES AND LABELS USED IN THE PROGRAM, THE LINES ON WHICH THEY WERE USED, AND HOW THEY WERE USED. IF NEEDED, THE VARIABLES FROM SPECIFED FORTRAN SOURCE FILES CAN BE SAVED ALONG WITH THOSE OF OTHER FORTRAN SOURCE FILES AND LATER PRINTED OUT AS A SUPER INDEX GIVEING VARIABLE NAME AND THE NAMES OF ALL THE FORTRAN SOURCE MODULES IT WAS USED IN. ALSO INCLUDED IS THE CAPABILITY TO EXCLUDE FROM THE INDEX LISTING ALL VARIABLES THAT ONLY APPEAR IN A FORTRAN SOURCE MODULE IN A COMMON BLOCK ,TYPE OR EQUIVALENCE DECLARATION BUT ARE NOT USED ELSEWARE(OR LIST ONLY THOSE IF WANTED). THE USER CAN ALSO LIST ONLY THOSE VARIABLES THAT ARE GLOBAL (DEFINED IN A COMMON BLOCK) OR THOSE THAT ARE LOCAL. A VARIATION OF THE SUPER INDEX IS AVAILABLE TO LIST (ALMOST)ALL ENTRY POINTS, THE MODULES THAT THEY ARE DEFINED IN, AND ALL THE ENTRY POINTS THAT CALL THEM AND THAT THEY CALL.

INDEX WILL HANDLE ALL SOURCE FILES WRITTEN IN FORTRAN FOUR AND FORTRAN FOUR PLUS IN ACORDANCE WITH 'PDP11 FORTRAN LANGUAGE REFERENCE MANUAL' (DEC-11-LFLRA-C-D) AND THE FORTRAN 1977 STANDARD.

INDEX CAN BE GENERATED TO RUN UNDER THE RT-11 OR RSX-11M OPERATEING SYSTEMS.

```
README.1ST     3  29-Oct-80     INDEX .DOC    38  29-Oct-80
INDEX .HLP     3  29-Oct-80     INDEX .COM     2  29-Oct-80
IDXTKB.CMD     2  29-Oct-80     INDEX .CMD     2  29-Oct-80
LINETP.MAC    10  29-Oct-80     INDEX .MAC     9  29-Oct-80
BUFFER.MAC     5  29-Oct-80     IOLINE.MAC    16  29-Oct-80
FLOW  .MAC    33  29-Oct-80     OPNCLO.MAC    25  29-Oct-80
RSX   .MAC     1  29-Oct-80     STORE .MAC    10  29-Oct-80
GET   .MAC    15  29-Oct-80     IOCHR .MAC     5  29-Oct-80
IMPURE.MAC     5  29-Oct-80     RAD50 .MAC     3  29-Oct-80
OUTPUT.MAC    12  29-Oct-80     TABLE .MAC     5  29-Oct-80
EVAL  .MAC    37  29-Oct-80     SUPER .MAC    16  29-Oct-80
 22 Files, 257 Blocks
```

****

Roger L. Matus
First National Bank of Chicago
1 First National Plaza
Chicago, IL 60670
312-732-6996

TAPER is a general purpose reader of 800 b.p.i. magnetic tapes. Its specialty is to take character data in an unknown format and, through interaction with the user, write the data in Digital Equipment format to any write-enabled disc. TAPER may be used to read 'foreign' tapes.

TECHNICAL INFORMATION

The user interface and formatting is handled by a program written in FORTRAN. The tape handler is written in MACRO-11. Only the FORTRAN programs, the EBCDIC collating sequence file, and documentation with the nam TAPER was written by the First National Bank of Chicago. All EXTMT routines were obtained from the Digital Equipment Computer User's Society. EXTMT was written by N. A. Bourgeois of Scandia Laboratories and operated there for the United States Department of Energy.

```
 05-Nov-80
TAPER .DOC    14  30-Oct-80     TAPER .HLP    16  30-Oct-80
TAPER .FOR    13  30-Oct-80     TAPSUB.FOR     9  30-Oct-80
TAPER .OBJ    16  30-Oct-80     TAPSUB.OBJ    16  30-Oct-80
TAPER .SAV    52  30-Oct-80     EXTMT .DOC     3  30-Oct-80
EXTMT .MAC    23  30-Oct-80     EXTMT .OBJ     2  30-Oct-80
 10 Files, 164 Blocks
```

****

Lawrence L. Morton
Lawrence L. Morton, EE & Associates
1747 So. Douglass Rd.   Suite D
Anaheim, CA 92806
(714) 634-1662

A FORTRAN IV/RT-11 Program that will renumber statement lines in a FORTRAN IV source program. (RT-11, V3B)

```
LOOKUP.FOR     5  23-Oct-80     LOOKUP.OBJ    10  23-Oct-80
NCLOSE.FOR     1  23-Oct-80     NCLOSE.OBJ     4  23-Oct-80
NREAD .FOR     2  23-Oct-80     NREAD .OBJ     5  23-Oct-80
NIF   .FOR     2  23-Oct-80     NIF   .OBJ     6  23-Oct-80
NGOTO .FOR     2  23-Oct-80     NGOTO .OBJ     6  23-Oct-80
NACCEP.FOR     2  23-Oct-80     NACCEP.OBJ     4  23-Oct-80
LARRY1.OBJ    38  15-Nov-80     RESEQ .SAV    48  15-Nov-80
RESEQ .FOR    17  15-Nov-80     RESEQ .OBJ    25  15-Nov-80
 16 Files, 177 Blocks
```

333 Cedar Street
New Haven, CT 06150
203-436-3617

Utility routines for Fortran/RT11.

```
 05-Nov-80
DSLIB .DOC     56  30-Oct-80      DSLIB .FOR     18  30-Oct-80
ASLOOK.MAC      6  30-Oct-80      DSLIB .COM      1  30-Oct-80
 4 Files, 81 Blocks
```

****

Mr. Robert L. Sharp
7073 Jasper Drive
Middletown, MD 21769

A Polish Interpreter.

```
POLISH.MAC     83  02-Nov-80
 1 Files, 83 Blocks
```

****

Hal Hovland
Hal Systems Corp
3544 SW 172nd
Seattle, WA 98166
(206) 244-6606

        FORTRAN callable macros to:
Move word, Move byte, ASCII/integer conversions, shift bits.

```
HALIB .MAC     20  03-Nov-80
 1 Files, 20 Blocks
```

****

Robert Abramson
Digital Equipment Corp.
110 Spit Brook Road
Nashua, NH 03061
603-884-8333

Summary description of PDP-11 FORTRAN OTS

```
README.DAT      1  27-Oct-80      F4OTS .DOC    439  27-Oct-80
 2 Files, 440 Blocks
```

****

David Stagg
Yale Medical School Dep't Pharmacology

****

Robert B. Denny
Creative System Design
3452 E. Foothill Blvd. Suite 601
Pasadena, Ca. 91107
(213) 792-9474

Complete distribution of DECUS Standard RUNOFF, Version M02.2 Includes RT-11 support, as well as RSX-11M, IAS and RSTS/E. This is the latest version, with many enhancements and bug fixes from older versions. It is the Text Processing SIG's standard version, previously not available for RT-11. Has RT-11 .SAV file, ready to run.

```
RNOASM.CMD    RNOBLD.CMD    RNOIAS.CMD    RSTASM.CMD
RSTBLD.CMD    RNOASM.COM    RNOLNK.COM    RNOBLD.CTL
RUNOFF.DOC    CMTAB .MAC    COMND .MAC    ERMSG .MAC
FMTCM .MAC    HYPHEN.MAC    INDEX .MAC    PINDX .MAC
RNCMD .MAC    RNFIO .MAC    RNORSX.MAC    RNPRE .MAC
RNRT11.MAC    RSTS  .MAC    RT11  .MAC    RUNOFF.MAC
START .MAC    UARITH.MAC    SMAC  .MLS    RBLDFC.ODL
RNO   .ODL    RNOBLD.ODL    RNOIAS.ODL    RUNOFF.RNO
RUNOFF.SAV
```

RT-11 conversion of David Sykes' excellent RATFOR preprocessor. This is a slightly earlier version than is in use on RSX-11M, but it is Just fine for RT-11. No RT-specific documentation (sorry), read the comments at the head of RATRT.RAT for RT-11 usage information. Also has nice runtime library. OK for RT-11 FORTRAN V2.5 on inline code. Malfunctions on earlier FORTRAN V2's in inline code.

```
              DIST  .COM    RATBLD.COM    GETSTR.FOR
PUTSTR.FOR    RATRT .FOR    RAT1  .FOR    RAT2  .FOR
RAT3  .FOR    STRLIB.FOR    RATHLP.MAC    RATFOR.MAP
STRLIB.OBJ    CCHAR .RAT    CDATIM.RAT    CDEFIO.RAT
CFOR  .RAT    CFUNC .RAT    CLINE .RAT    CLIST .RAT
CLOOK .RAT    COUTLN.RAT    CPRTLN.RAT    CSTR  .RAT
CUCLC .RAT    DEFIN .RAT    DEFINS.RAT    RATDEF.RAT
RATRT .RAT    RAT1  .RAT    RAT2  .RAT    RAT3  .RAT
STRLIB.RAT    TEST1 .RAT    TEST2 .RAT    TEST3 .RAT
TEST4 .RAT    TEST5 .RAT    CFILES.RT     RATFOR.SAV
RATFOR.DOC    RATFOR.RNO
```

UNOFFICIAL versions of DECUS 'C' system. Submitted to whet your appetite. The full official source distribution is available from the Structured Languages SIG and the DECUS Library. Look at XRF.COM for usage example. Implements most UNIX library funct's per Kernighan and Ritchie. Put *.(SAV,OBJ) and STDIO.H on a disk and assign it as device "C:" before use. The documentation included is PRELIMINARY also!!! CAVEAT!!! If you're at all interested in this, GET THE OFFICIAL TAPE.

```
                            CC    .SAV    AS    .SAV
 XR    .SAV    SUPORT.OBJ    CLIB  .OBJ    STDIO .H
 XRF   .H      XRFI  .C      XRFD  .C      XRFO  .C
 XRF1  .C      XRF2  .C      XRF3  .C      XRF   .COM
 CSYST .DOC    CC    .DOC
```

****

```
SUBMIT.DOC     6  02-Nov-80      RNOASM.CMD     2  02-Nov-80
RNOBLD.CMD     2  02-Nov-80      RNOIAS.CMD     3  02-Nov-80
RSTASM.CMD     1  02-Nov-80      RSTBLD.CMD     2  02-Nov-80
RNOASM.COM     1  02-Nov-80      RNOLNK.COM     1  02-Nov-80
RNOBLD.CTL     4  02-Nov-80      RUNOFF.DOC   146  02-Nov-80
CMTAB .MAC    12  02-Nov-80      COMND .MAC     8  02-Nov-80
ERMSG .MAC    10  02-Nov-80      FMTCM .MAC    26  02-Nov-80
HYPHEN.MAC    55  02-Nov-80      INDEX .MAC    11  02-Nov-80
PINDX .MAC     9  02-Nov-80      RNCMD .MAC    21  02-Nov-80
RNFIO .MAC     9  02-Nov-80      RNORSX.MAC     6  02-Nov-80
RNPRE .MAC     5  02-Nov-80      RNRT11.MAC    30  02-Nov-80
RSTS  .MAC     1  02-Nov-80      RT11  .MAC     1  02-Nov-80
RUNOFF.MAC    74  02-Nov-80      START .MAC     9  02-Nov-80
UARITH.MAC     3  02-Nov-80      SMAC  .MLS     6  02-Nov-80
RBLDFC.ODL     2  02-Nov-80      RNO   .ODL     1  02-Nov-80
RNOBLD.ODL     2  02-Nov-80      RNOIAS.ODL     1  02-Nov-80
RUNOFF.RNO   122  02-Nov-80      RUNOFF.SAV    33  02-Nov-80
DIST  .COM     3  02-Nov-80      RATBLD.COM     3  02-Nov-80
GETSTR.FOR     2  02-Nov-80      PUTSTR.FOR     2  02-Nov-80
RATRT .FOR    65  02-Nov-80      RAT1  .FOR    43  02-Nov-80
RAT2  .FOR    41  02-Nov-80      RAT3  .FOR    49  02-Nov-80
STRLIB.FOR    70  02-Nov-80      RATHLP.MAC     5  02-Nov-80
RATFOR.MAP    16  02-Nov-80      STRLIB.OBJ    95  02-Nov-80
CCHAR .RAT     2  02-Nov-80      CDATIM.RAT     1  02-Nov-80
CDEFIO.RAT     1  02-Nov-80      CFOR  .RAT     1  02-Nov-80
CFUNC .RAT     1  02-Nov-80      CLINE .RAT     1  02-Nov-80
CLIST .RAT     2  02-Nov-80      CLOOK .RAT     1  02-Nov-80
COUTLN.RAT     1  02-Nov-80      CPRTLN.RAT     1  02-Nov-80
CSTR  .RAT     1  02-Nov-80      CUCLC .RAT     1  02-Nov-80
DEFIN .RAT     6  02-Nov-80      DEFINS.RAT     4  02-Nov-80
RATDEF.RAT     6  02-Nov-80      RATRT .RAT    54  02-Nov-80
RAT1  .RAT    34  02-Nov-80      RAT2  .RAT    32  02-Nov-80
RAT3  .RAT    38  02-Nov-80      STRLIB.RAT    61  02-Nov-80
TEST1 .RAT     1  02-Nov-80      TEST2 .RAT     1  02-Nov-80
TEST3 .RAT     2  02-Nov-80      TEST4 .RAT     1  02-Nov-80
TEST5 .RAT     1  02-Nov-80      CFILES.RT      2  02-Nov-80
RATFOR.SAV    77  02-Nov-80      RATFOR.DOC   104  02-Nov-80
RATFOR.RNO    95  02-Nov-80      CC    .SAV    97  02-Nov-80
AS    .SAV    46  02-Nov-80      XR    .SAV    23  02-Nov-80
SUPORT.OBJ     1  02-Nov-80      CLIB  .OBJ    76  02-Nov-80
STDIO .H       5  02-Nov-80      XRF   .H       4  02-Nov-80
XRFI  .C       9  02-Nov-80      XRFD  .C       2  02-Nov-80
XRFO  .C      12  02-Nov-80      XRF1  .C      12  02-Nov-80
XRF2  .C      11  02-Nov-80      XRF3  .C       7  02-Nov-80
XRF   .COM     1  02-Nov-80      CSYSTM.DOC   466  02-Nov-80
CC    .DOC   109  02-Nov-80
 91 Files, 2433 Blocks
```

****

Greg L. Adams
Dept. of national Defence
National Defence Headquarters
Attn: DACS 2-2-4
Ottowa, Canada K1A0K4
(613) 993-9624

SPOOL/RT-11 V1.01, AUTOQ-automatic queing utility. Based on the paper "A Transparent Spooler for RT-11" given in Chicago 80, SPOOL/RT-11 V1.01 is an updated version which includes documentation.

DSC/RT-11 is a verifying disk scanner which reports the unique and common files found on the file structured devices (disk & mag tape).

```
SPRELN.DOC    13  17-Oct-80      SPUSER.DOC    37  17-Oct-80
SP    .MAC    11  17-Oct-80      SPV3B .MAC    13  17-Oct-80
SPOOL .MAC     9  20-Apr-80      SPINT .MAC    26  23-Aug-80
SPINIT.MAC    25  17-Oct-80      SPOOL .REL     8  17-Oct-80
SPOOLX.REL     7  17-Oct-80      SP    .SYS     2  17-Oct-80
SPX   .SYS     2  17-Oct-80      SPBLD .COM     1  17-Oct-80
SPBLDX.COM     1  17-Oct-80      SPBLDL.COM     1  17-Oct-80
DSC   .SAV    24  01-Nov-80      DSC   .RNO    14  01-Nov-80
DSC   .DOC    19  01-Nov-80      DSCLIN.PAS     3  28-Aug-80
DSCASC.PAS     3  21-Aug-80      DSCCMP.PAS     4  27-Aug-80
DSCPRN.PAS     4  28-Aug-80      DSCCSI.PAS    12  01-Nov-80
DSCMAN.PAS     5  01-Nov-80      DSCHED.PAS     4  01-Nov-80
DSCSET.PAS     9  01-Nov-80      DSCCMP.COM     1  15-Sep-80
DSCTST.COM     1  25-Aug-80      DSCLNK.COM     1  27-Aug-80
DSCMAC.COM     1  31-Oct-80      README.TXT     2  01-Nov-80
AUTOQ .OBJ     3  22-Mar-80      AUTOQ .REL    11  16-Mar-80
AUTOQ .MAC    14  22-Mar-80
33 Files, 291 Blocks
```

****

George G. Preckshot
University of Arizona
Tucson, AZ 85721
(602) 626-3450

Miscellaneous Handlers and formatters.

```
PH    .MAC    33  05-Nov-80      LPTT40.MAC    21  05-Nov-80
HS    .MAC    26  05-Nov-80      CLEAN .MAC    12  05-Nov-80
LPOUT .FOR     5  05-Nov-80      STRIP .MAC     2  05-Nov-80
PH    .DOC     3  05-Nov-80      README.DOC     7  05-Nov-80
CLEAN .DOC     4  05-Nov-80      LPOUT .DOC     5  05-Nov-80
LPTT40.DOC     8  05-Nov-80      HS    .DOC    16  05-Nov-80
 12 Files, 142 Blocks
```

See the next issue of this newsletter for information as to how you may obtain this RT-11 Fall '80 Symposium Tape.

SYSCON DESIGN INC.

815 MANHATTAN AVE. Suite F
MANHATTAN BEACH, CA. 90266
(213) 372-8544

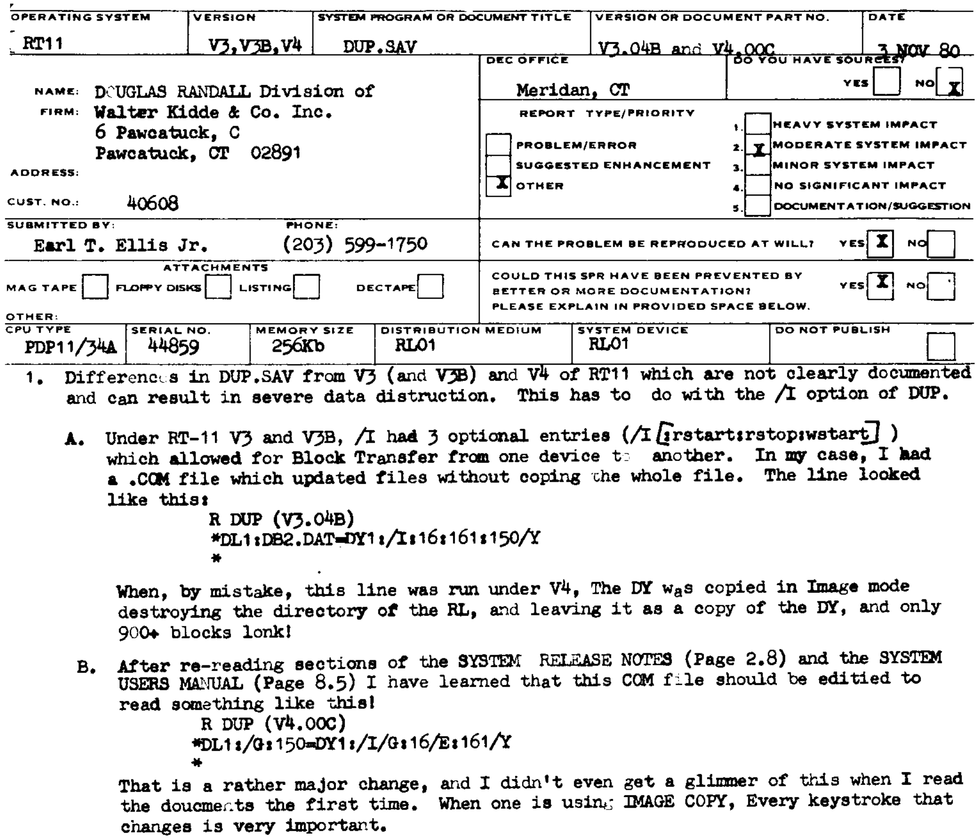
FMX - File Management System
GP-10, DL100 Graphics Software

FMX is a disk-based package of FORTRAN-callable routines which provide an ISAM (Indexed Sequential-Access Method) file capability to the FORTRAN programmer. The multilevel directory structure of FMX enables the number of disk accesses per data record retrieved to be cut to the absolute minimum while allowing very efficient allocation of disk space. Data records consist of key fields and data fields, where the data fields may be variable length. FMX supports random, sequential, and mixed access modes, and provides special routines for queue maintenance and partitioned files.

GP-10 is a package of FORTRAN-callable subroutines which provide a basic yet versatile graphics capability to the FORTRAN programmer. GP-10 will operate any graphics device accepting the Tektronix character codes, which include the Lear-Siegler ADM-3A with a Retro-Graphics board, and the Retro-Graphics VT-100. The basic MOVE, DRAW, and POINT routines are available in device and virtual (user-defined) coordinates, and in absolute and relative modes. In addition, there are utilities for displaying text, defining dashed lines, and doing interactive graphics with such features as cursor control and selective erasure.

DL-100 is an interactive graphics program which enables the user to construct and annotate graphs of data stored on a computer file using English-like commands. Data files may be entered manually or generated by programs written in any language. The user may choose from a number of plot types, define axes, define symbols to highlight plotted points, and display text horizontally and vertically.

FMX, GP-10, and DL100 are currently available for DEC PDP-11 computers using RT-11 and TSX operating systems. For more information contact SYSCON DESIGN INC., 815 Manhattan Avenue, Suite F, Manhattan Beach, CA 90266; telephone (213) 372-8544.



OPERATING SYSTEM	VERSION	SYSTEM PROGRAM OR DOCUMENT TITLE	VERSION OR DOCUMENT PART NO.	DATE
RT11	V3,V3B,V4	DUP.SAV	V3.04B and V4.00C	3 NOV 80

NAME: DOUGLAS RANDALL Division of
FIRM: Walter Kidde & Co. Inc.
6 Pawcatuck, C
Pawcatuck, CT 02891
ADDRESS:
CUST. NO.: 40608

DEC OFFICE: Meridan, CT

DO YOU HAVE SOURCES? YES □ NO ☒

REPORT TYPE/PRIORITY
□ PROBLEM/ERROR
□ SUGGESTED ENHANCEMENT
☒ OTHER

1. □ HEAVY SYSTEM IMPACT
2. ☒ MODERATE SYSTEM IMPACT
3. □ MINOR SYSTEM IMPACT
4. □ NO SIGNIFICANT IMPACT
5. □ DOCUMENTATION/SUGGESTION

SUBMITTED BY: Earl T. Ellis Jr. PHONE: (203) 599-1750

CAN THE PROBLEM BE REPRODUCED AT WILL? YES ☒ NO □

ATTACHMENTS
MAG TAPE □ FLOPPY DISKS □ LISTING □ DECTAPE □
OTHER:

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES ☒ NO □

CPU TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE	DO NOT PUBLISH
PDP11/34A	44859	256Kb	RL01	RL01	□

1. Differences in DUP.SAV from V3 (and V3B) and V4 of RT11 which are not clearly documented and can result in severe data distruction. This has to do with the /I option of DUP.

   A. Under RT-11 V3 and V3B, /I had 3 optional entries (/I [:rstart:rstop:wstart]) which allowed for Block Transfer from one device to another. In my case, I had a .COM file which updated files without coping the whole file. The line looked like this:

   ```
   R DUP (V3.04B)
   *DL1:DB2.DAT=DY1:/I:16:161:150/Y
   *
   ```

   When, by mistake, this line was run under V4, The DY was copied in Image mode destroying the directory of the RL, and leaving it as a copy of the DY, and only 900+ blocks lonk!

   B. After re-reading sections of the SYSTEM RELEASE NOTES (Page 2.8) and the SYSTEM USERS MANUAL (Page 8.5) I have learned that this COM file should be editied to read something like this!

   ```
    R DUP (V4.00C)
   *DL1:/G:150=DY1:/I/G:16/E:161/Y
   *
   ```

   That is a rather major change, and I didn't even get a glimmer of this when I read the doucments the first time. When one is using IMAGE COPY, Every keystroke that changes is very important.

CSIRO

Division of Textile Physics
Ryde, NSW 2112

A Division of the Institute of Industrial Technology

RNC:CR
Our Ref: PH/CAF.003/E

338 Blaxland Road, Ryde. NSW 2112
Telephone (02) 80 0211
Telegrams Textres, Sydney Telex 20248

SPR Section. Fortran IV V2.1 Inline Code

In the previous issue of Mini-tasker the problem of the Inline code from the V2.1 compiler was raised by Mark Liverman. I have identified the problem and submitted an SPR. The reply I received was, however, rather common: "... the next release ..." etc. I enclose a copy of the SPR response.

Put simply, the problem of faulty code generation by the Inline section of the compiler is a deep rooted one of poor design in the register optimisation phase. The failure is quite reproducible. Given any situation where two integer variables are in use close together, and in particular if they are used as array indices, the optimiser may attempt to keep them both in registers rather than in memory. This is arguably an undesirable practice in a real-time Fortran system with user-written interrupt-driven processes going on. Anyhow, the optimiser can fail and get the variables tangled. Of course, if you jumble the statements around a bit, you may be able to fool the optimiser into not trying. I have found the following line to be an excellent example of this:

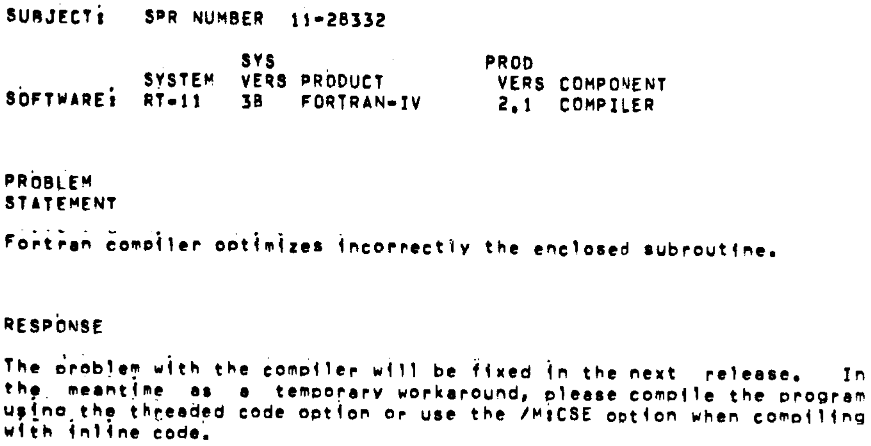
```
Type *, I, J, A(I,J).
```

With optimisation for space, this will be wrongly compiled and will produce quite erroneous output for A(I,J) at least. Examination of the generated code usually shows RØ and R1 being confused.

I am afraid that the only safe answer is to go back to using threaded code. The drawback is a small loss of speed. The advantages are reliable code, smaller core images for large programs, more error checking, a smaller compiler, etc. Alternatively, as noted in the SPR reply, try using /M:CSE with inline code. This has the effect of turning off the optimiser section concerned with common subexpressions.

Yours sincerely,

R.N. Caffin



SUBJECT: SPR NUMBER 11-28332

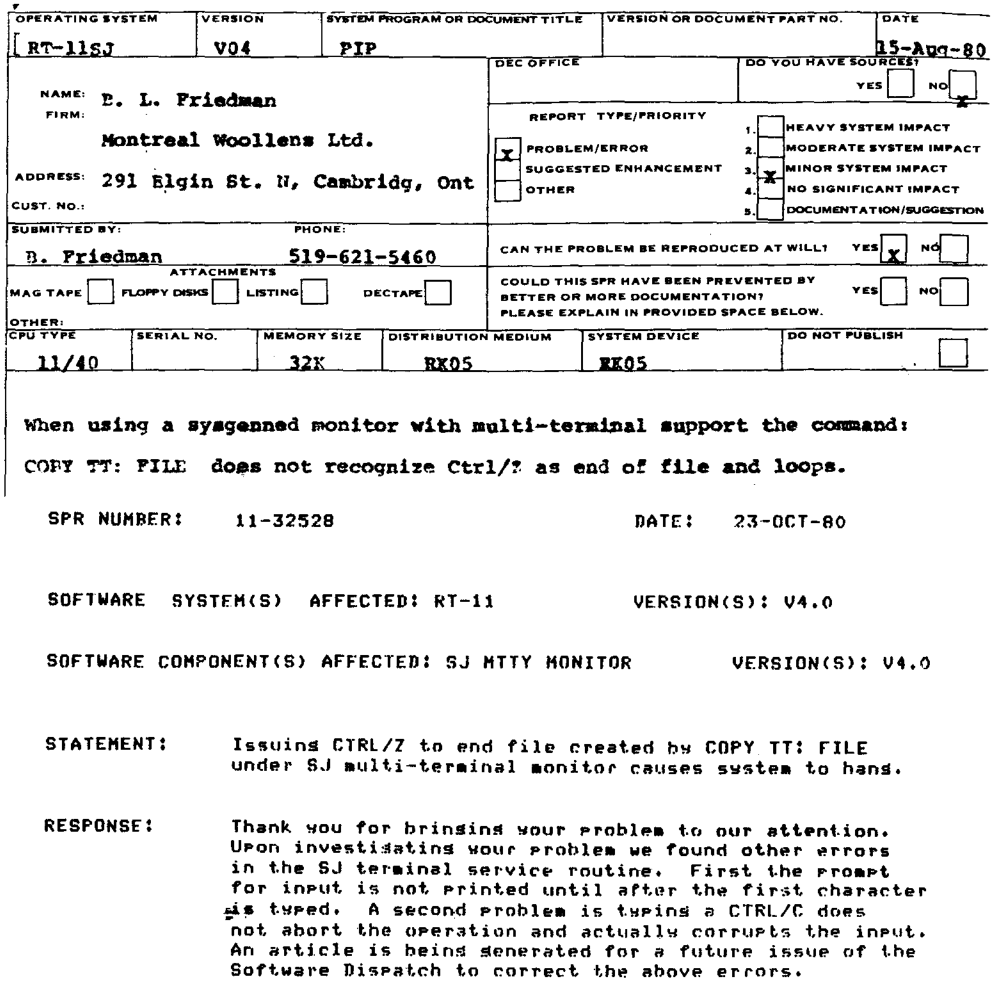
	SYSTEM	SYS VERS	PRODUCT	PROD VERS	COMPONENT
SOFTWARE:	RT-11	3B	FORTRAN-IV	2.1	COMPILER

PROBLEM STATEMENT

Fortran compiler optimizes incorrectly the enclosed subroutine.

RESPONSE

The problem with the compiler will be fixed in the next release. In the meantime as a temporary workaround, please compile the program using the threaded code option or use the /M:CSE option when compiling with inline code.



OPERATING SYSTEM	VERSION	SYSTEM PROGRAM OR DOCUMENT TITLE	VERSION OR DOCUMENT PART NO.	DATE
RT-11SJ	V04	PIP		15-Aug-80

NAME: E. L. Friedman
FIRM: Montreal Woollens Ltd.
ADDRESS: 291 Elgin St. N, Cambridg, Ont
CUST. NO.:

DEC OFFICE

DO YOU HAVE SOURCES? YES □ NO ☒

REPORT TYPE/PRIORITY
☒ PROBLEM/ERROR
□ SUGGESTED ENHANCEMENT
□ OTHER

1. □ HEAVY SYSTEM IMPACT
2. □ MODERATE SYSTEM IMPACT
3. ☒ MINOR SYSTEM IMPACT
4. □ NO SIGNIFICANT IMPACT
5. □ DOCUMENTATION/SUGGESTION

SUBMITTED BY: B. Friedman PHONE: 519-621-5460

CAN THE PROBLEM BE REPRODUCED AT WILL? YES ☒ NO □

ATTACHMENTS
MAG TAPE □ FLOPPY DISKS □ LISTING □ DECTAPE □
OTHER:

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES □ NO □

CPU TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE	DO NOT PUBLISH
11/40		32K	RK05	RK05	□

When using a sysgenned monitor with multi-terminal support the command:

COPY TT: FILE does not recognize Ctrl/Z as end of file and loops.

SPR NUMBER: 11-32528 DATE: 23-OCT-80

SOFTWARE SYSTEM(S) AFFECTED: RT-11 VERSION(S): V4.0

SOFTWARE COMPONENT(S) AFFECTED: SJ MTTY MONITOR VERSION(S): V4.0

STATEMENT: Issuing CTRL/Z to end file created by COPY TT: FILE under SJ multi-terminal monitor causes system to hang.

RESPONSE: Thank you for bringing your problem to our attention. Upon investigating your problem we found other errors in the SJ terminal service routine. First the prompt for input is not printed until after the first character is typed. A second problem is typing a CTRL/C does not abort the operation and actually corrupts the input. An article is being generated for a future issue of the Software Dispatch to correct the above errors.

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

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.