THE minitasker

DECUS
RT-11 SIG NEWSLETTER

APRIL 1980 VOL. 6 NO. 2

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 may 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

The recently completed Chicago Symposium contained a wide range of useful information for RT-11 users. The next issue of the 'minitasker' will feature various Symposium sessions.

User Input

Dear Mr. Demers:

After having to recover data from three different bad disks in two weeks, the thought occurred to me that some of your readers may also want to know how to recover data from a corrupted disk. But just a warning: Do not try to recover data from a disk which has been involved in a "head crash.":

HOW A HARD DISK IS MADE

A hard disk is made from a carefully machined aluminum platter which has been coated on both sides with a magnetic coating. When the disk pack is put into a drive and it is spinning with the heads loaded, the heads are suspended above the rotating disk on a thin film of air. Only a few millionths of an inch of air separate the heads from the disk surface. As forgiving as disks often are, a particle of dust, smoke, or even a fingerprint on the surface of the disk can cause a colision with a head of the disk drive. Since the disk at its outside edge is traveling at about 1000 inches per second, such a minor collision can warp the head so that it makes physical contact with the surface of the disk. This can cause the head to actually plow a neat little furrow around the disk platter, spewing flakes of the magnetic coating and aluminum shavings throughout the inside of the disk drive.

The damage from head crash can be considerable. Just picture the mechanics and electronics of the inside of a drive throughly coated with sparkling, conducting chips of aluminum. This tends to be hard on both the drive and on the data on the disk pack. (The description comes from first hand experience -- I have a drive which had a head crash about three years ago. We vacuumed out the drive and dismantled and vacuumed and replaced the heads and vacuumed and crossed our fingers, put an old pack in it and it worked! And it is still working as the main disk drive in my #2 computer system -- running constantly ever since. But whenever we open it up to clean the heads, the insides still glitter like a Las Vegas chorus line.

And I think we've all heard the story about the fellow who had a disk so bad, so he put in another pack, but it didn't work either, so he put that pack in another drive, ad nauseam, until all of his drives had head crashes because of the crashed pack he kept moving around, and of course these newly crashed drives destroyed other disk packs wiping out all of his backup packs ... let's hear it for mag tape.

Anyway, while I am telling you how to recover data from a corrupted pack, keep in mind that you should never try to read physically damaged media lest you damage the drive also and any other media which follows its way into the drive.

Well, then, what is corrupted media which can be read?

A. Disks with corrupted directories.

B. Disks with hard errors.

C. Deleted files.

D. Stupid errors ("REPLACE RK0:" in Basic).

Some of these can be recovered easier and more reliably than others. But let's explain a few things first.

1. Formatting a disk (completely) results in a disk with no recoverable data on it whatsoever.

2. A partially formatted disk (you know, one of those late night sessions when you tell the machine to format the wrong disk and have the immediate presence-of-mind to turn off the power on the computer or halt it before it stops formatting. Control C's just don't cut it in this situation.) has recoverable data on the portion of the disk which wasn't just freshly formatted.

3. A zeroed directory (INIT) does just that ... it zeroes the directory. The data is still on the disk -- only the directory has been erased. Given enough patience and a long enough lifespan, the entire disk can be recovered.

THE TOOLS

RT-11 provides several tools which are commonly used to operate on files. Some of these can be used to also read and write a device -- not just a file on the device.

My favorites are (in order of preference):

TECO to find, read, and transfer.

BASIC to find, read, and transfer. (I guess that FORTRAN will too, but it isn't interactive like TECO or BASIC).

DUMP -- to find -- especially RAD50.

PATCH -- to try to correct -- of limited usefulness generally, but in specific cases extremely valuable.

ODT - Mostly useful by first-class Wizards or better. For getting it right from the horse's mouth. Careless keystrokes are sometimes dangerous.

THE PROCEDURE:

1) Copy the media to a freshly formatted like-media or a larger media (copy a bad floppy onto an RK if you need to, or a bad RK onto an RL) using the COPY/DEVICE command. (If you are a wizard second-class and you only have two drives, use ODT microcode on an LSI (or on any other machine BOOT RT-11 from a good pack, SET USR NOSWAP, RUN ODT.SAV (JUST LINK ODT.OBJ by itself) and then remove your system pack). Put in a scratch pack and key in a format program if necessary using ODT (one

of the RT11-V2C Books had a format program in it -- I think it was the System Reference Manual) and then key in a program to read a sector and write a sector and just spin on the ready bit.) But it is much easier if you have 3 drives (a System drive (even floppy or TU58 will do) and the corrupted disk and the new copy).

2) After you have a copy of the corrupted disk, you can breathe a little easier and put a minimum-system disk in one drive and the copy of the bad disk in the other. The minimum-system disk should have at least: RKMNXX.SYS, SWAP.SYS, RK.SYS, TT.SYS, LP.SYS, NL.SYS, TECO.SAV, DUP.SAV, DIR.SAV, PIP.SAV, PATCH.SAV, BASIC.SAV,ODT.SAV, DUMP.SAV, and SRCCOM.SAV.

3)If you know exactly what you want to recover: (for example you backed up the disk yesterday, but you really need to see the latest copy of the program since then and was stored on the bad disk), assuming it began with '.TITLE FOO':

```
R TECO
*ERRK0:$$(the bad disk)
*_.TITLE FOO
$-T$$
```

and wait. (You will encounter several '.TITLE FOO's on your disk because old data is never erased from a disk -- even when it is DELETEd -- it is just written over again and again. Your goal is to find the latest copy.) Sooner or later, TECO will respond with:

```
 .TITLE FOO
*
(Then type)
-L0,.K$<A>$$
(TECO responds with)
MEMORY OVERFLOW
(then type)
JS.ENDS$$
```

(The 'S' on the end keeps you from landing on '.ENDC', '.ENDM', or '.ENDR').

```
*LHT$$
```

(And watch the world go by. Adjust as necessary.)

```
*EWRK2:FOO1.MAC$0,.PWEF$$
*(Do the _.TITLE FOO
```

$$ again and copy it out onto 'RK2:FOO2.MAC', etc until you get an input error.

An input error on TECO indicates an end-of-volume or a hard error. TECO stops and will not cross a hard error, but a COPY/DEVICE will -- that is one of the reasons for the COPY/DEVICE -- to have a volume free of hard errors.

4) If you haven't backed up the volume for a couple of decades and you want to try to recover everything, may I suggest BASIC?

```
5 L% = (however-many-characters at the start of each block you want to
 see)
10 DIM  1, A$(4800,3)=128 \ REM THE 4800 IS THE NUMBER OF BLOCKS IN AN
 RK
20 OPEN 'RK0:' FOR INPUT AS FILE  1
30 FOR A% = 0 TO 4800
40 A$ =A$(A%,0)
```

```
50 A1$=SEG$(A$,1,LX)
60 PRINT AX,A1%
70 NEXT A%
80 CLOSE  1
90 END
```

This works best on an LA120 where the page length can be set to one line because many FORM FEEDS will undoubtedly be encountered and using this program to print out 4800 lines wastes enough paper as it is.

Once the block numbers to be saved have been determined, use:

```
 .R DUP
 *FILE1.TST=/C:(start block number).:(file length).
 *FILE2.TST=/C: (... as above)
```

Read page 8-3 in the System User's Guide and don't forget to type a decimal point after the numbers. Also note the contiguous-file restrictions.

5) Patching a directory:

```
R PATCH
FILE NAME --- RK0:/A
*6000/ nnnnnn
```

Read section C.7.1 "RT-11 File Storage" beginning on page C-65 in the Advanced Programmer's Guide for information on what an octal representation of a directory looks like and what all the numbers mean.

There is an undocumented feature in PATCH which allows you to search for the storage location of any number in any file on any device. That feature is:

```
*nnnnnn;S  (where nnnnnn is the number you are searching for)
```

If you have used the '/A' switch on the file name, the ';S' command will cause a read error and PATCH will return to the monitor on encountering end-of-file. No harm has been done -- just run Patch again.

DUMP can be used to print out the contents of the directory.

```
R DUMP
*TT: (or LP:) =RK0:/S:6/E:20/X
```

where '6' is always the start of the directory on any device, and '20' is a pretty good guess about where most directories end. If you need more, then print more.

ODT.SAV can be used to make RAD50 to octal conversions.

```
R ODT

*400/000000   X= F
402/000000    X= ILE
404/000000    X= BAD
*400/000006
402/035045
404/006254
```


The purpose of this document has been to spur your thinking a little. To fill in all of the necessary details would take a whole book. And of course experience is the best teacher.

My final suggestion is that you try all of this first on a spare COPY of a scratch disk so that if you erase any information (by writing over it) there will be no consequences. For you who don't regularly use TECO and have not become intimate with its syntax and cryptic error messages, do try to start to use TECO as your regular editor. It is painful at first, but it and ODT.SAV (LINK ODT) are my favorite tools for getting out of bad trouble.

Sincerely,
Ray Strackbein

SUBMITTED BY: LARRY SEARS, ED SERVICES, LANHAM, MD 20706
[redacted] 301-459-7900 x2555

ONE CAN MODIFY THE FLOPPY HANDLER (DX.MAC)
TO INCLUDE A SET OPTION TO WRITE PROTECT THE FLOPPY.
IF IMPLEMENTED ONE WOULD BE ABLE TO WRITE PROTECT
THE FLOPPY DISKS AS LONG AS YOU ARE NOT BOOTED FROM THEM.
THE SET COMMAND MODIFIES DX.SYS NOT THE SYSTEM DEVICE HANDLER
IN RMON.
THE SET CMD YOU WILL BE ABLE TO USE WOULD BE:
SET DX: WP
SET DX: NOWP
THIS IS A PERMANENT CHANGE IN THE HANDLER.

BELOW IS THE CODE YOU CODE INCLUDE IN THE HANDLER:

- Not a DEC supported feature
- Could be used for RX02 also

```
;************ NEW CODE ***************
.ASECT          ;SET OPTION TABLE
.=400
        NOP     ;NOP INSTRUCTION PLACED INTO R3
        .RAD50 /WP    /
        .BYTE <O.WP-400>/2
        .BYTE 200       ;YES/NO MODE OF SET CMD -"SET DX WP" OR "SET DX NOWP"
        .WORD 0         ;ENDS THE LIST
O.WP:   MOV (PC)+,R3    ;ROUTINE TO MODIFY HANDLER -ENTER HERE IF YES
        BMI RXERRJ-WP+. ;!!!!PIC!!!!
        MOV R3,WP       ;ENTER HERE IF NO
        RTS PC
.PSECT
************** END OF NEW SET OPTION TABLE**********
```

```
.IF NE DXT$O
        .DRBEG  DX,DX$VEC,DXDSIZ,DXSTS,RXTVTB
.IFF
        .DRBEG  DX,DX$VEC,DXDSIZ,DXSTS
.ENDC
        MOV     #RETRY,(PC)+
RXTRY:  .WORD   0
        MOV     DXCQE,R3
;*********NEW LINES*****
        TST 6(R3)       ;TEST TO SEE IF WORD COUNT IS NEG(WRITING)
WP:     NOP             ;IF "SET DX WP" WILL CONTAIN A "BMI RXERRJ"
;******** END OF NEW LINES******
        MOV     (R3)+,R5
        MOV     #CSGO+CSRD,R4
        MOVB    (R3)+,R1

.IF NE DXT$O
        BITB    #4,@R3
        BNE     RXERRJ

.IF NE ERL$G
```

COMPARISON OF SEVERAL SORTING METHODS FOR NUMBERS OR CHARACTERS
RUNNING ON A PDP11/05 UNDER RT11 IN FORTRAN IV LANGUAGE

****

BY D. GUINIER AND R. KIRSCH

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

INTRODUCTION :
****

IT IS VERY CONVENIENT TO HAVE AN EFFICIENT METHOD TO SORT ARRAYS OF NUMBERS (INTEGERS OR REAL NUMBERS) OR CHAINS OF CHARACTERS.
OUR OBJECT WAS TO STUDY AND COMPARE THE PERFORMANCES OF SEVERAL SORTING METHODS TO LEARN WHICH IS THE MOST RAPID.

METHODS :
****

IN THIS WORK WE HAVE COMPARED FOUR METHODS :

1) THE PERMUTATION METHOD (PERM) DESCRIBED BY DAVIES R. G. (1971)
2) THE BINARY TREE METHOD (TREE)
3) THE MAC-LARREN METHOD (TERSIM) DESCRIBED BY MAC-LARREN D. (1966)
4) THE HIBBARD-HOARE-SINGLETON METHOD (SORT) DESCRIBED BY SCHAUB R. (1973).

RESULTS :
****

A) SORTING REAL*4 NUMBERS OF AN ARRAY :

ALL COMPARISONS OF THE FOUR METHODS ARE REPORTED IN THE IN THE TABLE AND THE GRAPH BELOW AFTER RANDOMISATION OF THE NUMBERS CORRESPONDING TO THE ITEMS.

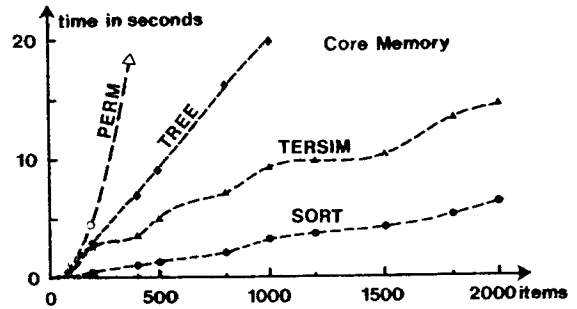
THE TABLE GIVE THE ELASPED TIME IN SECONDS TO SORT 100 TO 2000 ITEMS.

NUMBER OF ITEMS	PERM	TREE	TERSIM	SORT
100	2	1	0.5	
200	7	3	2.5	
400	30	7	3.5	1
500	48	9	5	1
800	122	16	7	2
1000	192	20	9	3
1200	279	24	9.5	3
1500	433	32	10	4
1800	624	40	13	5
2000	772	44	14	6

MEAN SPEED EXPRESSED IN MILLISECONDS BY ITEM SORTED.

SPEED	390	22	7	3

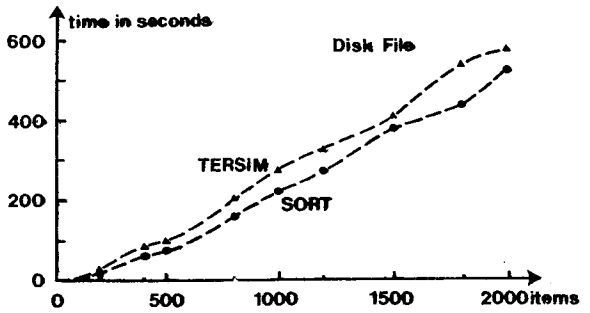
GRAPH OF THE ELASPED TIME AS A FUNCTION OF THE NUMBER OF ITEMS TO SORT.



CONCLUSION : THE SORT METHOD IS THE MOST EFFICIENT OF THE FOUR.

B) SORTING NUMERICAL RECORDS IN A RANDOM ACCESS RK05 DISK FILE :

THE GRAPH BELOW SHOWS THAT THE MANIPULATION OF NUMBERS CORRESPONDING TO RECORDS IN A RANDOM ACCESS RK05 DISK FILE UNDER RT11 INCREASES CONSIDERABLY THE ELASPED TIME BY FACTORS 40 OR 86 FOR THE TERSIM AND SORT METHOD, RESPECTIVELY.



CONCLUSION : IT'S NECESSARY TO TRANSFER THE RECORDS OF THE FILE IN AN ARRAY IN CORE MEMORY SO AS NOT TO INCREASE CONSIDERABLY THE SEARCH TIME. AT THE END OF THE OPERATIONS, THE RANKED ARRAY IS RECOPIED IN PLACE OF THE ORIGINAL FILE.

C) SORTING ASCII CHARACTERS FILE RECORDS BY THE SORT METHOD :

WE HAVE EMPLOYED THE FOLLOWING METHOD TO SPEED UP AND SIMPLIFY THE PREBLEM SET BY SORTING A FILE OF ALPHABETICAL OR ALPHANUMERICAL ITEMS :

A) FIRST WE CHANGED THE ASCII CHARACTERS OF THE ITEMS INTO RADIX50 REAL*4 (FOR 6 CHARACTERS) OR REAL*8 (FOR 12 CHARACTERS) VALUES IN ORDER TO BE ABLE TO ARRANGE REAL NUMBERS . AT THE END OF THE SORTING TREATMENTS OF THE REAL ARRAY, WE RESTORE THE RADIX50 REAL*4 OR REAL*8 VALUE INTO ASCII CHARACTERS CORRESPONDING TO EACH RECORD AND WRITE THESE SEQUENTIALY IN THE FILE.

THE SORTING SPEED IS ABOUT 3 MILLISECONDS/REAL*4 ITEM
THE SORTING SPEED IS ABOUT 4 MILLISECONDS/REAL*8 ITEM
THE SORTING SPEED IS ABOUT 2 MILLISECONDS/INTEGER ITEM

THE TOTAL ENCODING-DECODING PROCESS TIME IS 4 MILLISECONDS/ITEM
THE TOTAL SPEED OF SORTING FOR 12 CHARACTERS, INCLUDING UPDATING THE FILE IS ABOUT 14 MILLISECONDS/ITEM.

CONCLUSION :
****

-THE SORT METHOD PERMITS TO SORT INTEGER, REALS (SIMPLE OR DOUBLE PRECISION) OF SAME OR OPPOSITE SIGNS.
-THE ENCODING-DECODING PROCESS DESCRIBED ALLOWS THE SORTING OF ALPHABETICAL OR ALPHANUMERICAL RECORDS WITH A HIGH SPEED.

REFERENCES :
****

1) DAVIES R.G. (1971) : COMPUTER PROGRAMMING IN QUANTITATIVE BIOLOGY (ACADEMIC PRESS LONDON AND NEW YORK).
2) LOUIT G. (1971) : ALGORITHMES DE TRI (DUNOD PARIS).
3) MAC-LARREN D. (1966) : INTERNAL SORTING BY RADIX PLUS SIFTING (J.A.C.M.).
4) SCHAUB R. (1973) : ETUDE DE QUELQUES METHODES DE TRI (NOTE D'ETUDE AN-NO.35 C.C.S.A.).

LISTING : ENCODING-DECODING PROCESS AND SORT PROGRAM.
****

```
FORTRAN IV      V01C-03A

      C
      C-------------------------------------------------------------
      C  EXEMPLE OF USE.
      C-------------------------------------------------------------
      C
0001        DIMENSION A(2000),NN(2)
0002        REAL*8 A
0003        LOGICAL*1 LIGNE(12)
      C
      C  INTEGER A : IF A IS AN INTEGER
      C  REAL*4  A : IF A IS A REAL
      C  REAL*8  A : IF A IS A REAL DOUBLE PRECISION
      C
      C                   OR A LOGICAL*1 (ASCII) CONVERTED
      C                   TO A RADIX50 VALUE (REAL*4 OR REAL*8)
```

```
      C
      C*******************************************************
      C
      C  FILE DEFINITION.
0004        CALL FICH(7,1,2048,6,INDX)
0005        WRITE(7,300)
0006  300   FORMAT('$NUMBER OF ITEMS N= ')
0007        READ(5,400)N
0008  400   FORMAT(I5)
      C
      C*******************************************************
      C
      C  FILE CONVERSION : ENCODING A 12 ASCII CHARACTERS IN A
      C  RADIX50 REAL*8 VALUE.
0009        DO 2 I=1,N
0010        READ(1'I)LIGNE
0011  2     CALL IRAD50(12,LIGNE,A(I))
      C  RANGE A IN ASCENDING ORDER.
0012        CALL SORT(A,1,N,0)
      C  FILE RECONVERSION : DECODING A RADIX50 REAL*8 VALUE IN
      C   12 ASCII CHARACTERS.
0013        DO 3 I=1,N
0014        CALL R50ASC(12,A(I),LIGNE)
      C  UPDATING THE FILE.
0015  3     WRITE(1'I)LIGNE
0016        STOP
0017        END
```

```
FORTRAN IV      V01C-03A

0001        SUBROUTINE FICH(IMP,NLOGIC,NENR,MOTS,INDX)
      C
      C  DEFINITION OF A DIRECT ACCESS FILE.
      C
0002        WRITE(IMP,100)
0003  100   FORMAT('$FILE NAME : ')
0004        CALL ASSIGN(NLOGIC,'DK:',-1)
0005        DEFINE FILE NLOGIC(NENR,MOTS,U,INDX)
0006        RETURN
0007        END
```

```
FORTRAN IV      V01C-03A

0001        SUBROUTINE SORT(A,II,JJ,IND)
0002        DIMENSION A(2000),IU(20),IL(20)
      C
      C  INTEGER A,T,TT : IF A IS AN INTEGER
      C  REAL*4  A,T,TT : IF A IS A REAL
      C  REAL*8  A,T,TT : IF A IS A DOUBLE PRECISION NUMBER
      C
      C                  OR A LOGICAL*1 (ASCII) CONVERTED
      C                  TO A RADIX50 VALUE (REAL*4 OR REAL*8)
      C
0003        REAL*8 A,T,TT
0004        M=1
0005        I=II
0006        J=JJ
```



```
0007        IF(IND.EQ.0)GO TO 8
0009  1     IF(A(I))2,2,3
0010  2     I=I+1
0011        IF(I.EQ.J)GO TO 7
0013        GO TO 1
0014  3     T=A(I)
0015  4     IF(A(J))6,5,5
0016  5     J=J-1
0017        IF(J.EQ.I)GO TO 7
0019        GO TO 4
0020  6     A(I)=A(J)
0021        A(J)=T
0022        GO TO 1
0023  7     IU(M)=JJ
0024        IF(A(I).LT.0)I=I+1
0026        IL(M)=I
0027        J=I-1
0028        I=II
0029        M=M+1
0030  8     IF(I.GE.J)GO TO 70
0032  10    K=I
0033        IJ=(J+I)/2
0034        T=A(IJ)
0035        IF(A(I).LE.T)GO TO 20
0037        A(IJ)=A(I)
0038        A(I)=T
0039        T=A(IJ)
0040  20    L=J
0041        IF(A(J).GE.T)GO TO 40
0043        A(IJ)=A(J)
0044        A(J)=T
0045        T=A(IJ)
0046        IF(A(I).LE.T)GO TO 40
0048        A(IJ)=A(I)
0049        A(I)=T
0050        T=A(IJ)
0051        GO TO 40
0052  30    A(L)=A(K)
0053        A(K)=TT
0054  40    L=L-1
```

```
FORTRAN IV      V01C-03A

0055        IF(A(L).GT.T)GO TO 40
0057        TT=A(L)
0058  50    K=K+1
0059        IF(A(K).LT.T)GO TO 50
0061        IF(K.LE.L)GO TO 30
0063        IF(L-I.LE.J-K)GO TO 60
0065        IL(M)=I
0066        IU(M)=L
0067        I=K
0068        M=M+1
0069        GO TO 80
```

```
0070  60    IL(M)=K
0071        IU(M)=J
0072        J=L
0073        M=M+1
0074        GO TO 80
0075  70    M=M-1
0076        IF(M.EQ.0)RETURN
0078        I=IL(M)
0079        J=IU(M)
0080  80    IF(J-I.GE.11)GO TO 10
0082        IF(I.EQ.II)GO TO 8
0084        I=I-1
0085  90    I=I+1
0086        IF(I.EQ.J)GO TO 70
0088        T=A(I+1)
0089        IF(A(I).LE.T)GO TO 90
0091        K=I
0092  100   A(K+1)=A(K)
0093        K=K-1
0094        IF(T.LT.A(K))GO TO 100
0096        A(K+1)=T
0097        GO TO 90
0098        END
```

USER REQUESTS

We have several PDP-11/34's within SAENET, which run MU-BASIC Version 2 under RT-11 V3B (XM). As part of our "accounting system" we need access to the beginning address of each (logged in) user partition. Page 4-21 of the MU-BASIC-11/Rt-11 Users Guide shows how to access this address with a routine "GTUSPT"; but this only works if you have an FB monitor. Would anyone who has any experience with accessing user partitions in MU-BASIC Version 2 in an XM monitor please contact:

Lloyd Roberts,
SAENET,
P.O. Box 1,
INGLE FARM. S.A. 5098,
AUSTRALIA

Yours sincerely,

J.L. ROBERTS,
Applications Programmer

Gruppe für Rüstungsdienste
Groupement de l'armement
Aggruppamento dell'armamento

Telegramme: Rüstungsdienst
Telex: 32 225 a GRD CH

No. FAR/NYM 262
In der Antwort bitte angeben
Indiquer dans la réponse s. v. p.

Technical Division 9

Ken Demers
MS - 48
United Technologies Research Center
Silver Lane
East Hartford, Connecticut 06108
USA

Ihr Zeichen / Votre référence	Ihre Nachricht vom / Votre communication du	Rückfragen an / Appel au	
--	--	031 67 61 22	3000 Bern 22/Switzerland Stauffacherstrasse 65 April 21, 1980

Dear Mr. Demers

In the Mini-Tasker, Vol. 5 No 4, Sept. 1979, you mention on page 16 two techniques to access I/O page when running the XM version of MU BASIC.

I am very interested in these methods and would appreciate to get a copy of the relevant information.

Sincerely yours,

DEFENSE TECHNOLOGY AND
PROCUREMENT AGENCY
Technical Division 9
The head of section 9.2

J.P. Farine

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

Dear Mr. Demers:

An entry for User Requests:

1. Super Star Trek has been transported to RT-11 Fortran IV - at least, a binary copy has surfaced. Are the sources for the RT-11 version available. If so, from whom?

2. HAUNT is a PDP-10-based maze-solving game in the Adventure-Dungeon mold. Are the sources available, either in an RT-11 compatible language or the original? If so, from whom?

Thanks.

Yours truly,

Robert M. Supnik

DIGITAL EQUIPMENT CORPORATION, 146 MAIN STREET, MAYNARD, MASSACHUSETTS 01754
(617) 897-5111 TWX: 710-347-0212 TELEX: 948457

ROYAL PRINCE ALFRED HOSPITAL
MISSENDEN ROAD
CAMPERDOWN, N.S.W. 2050

TELEPHONE: 51 0444 Ext. .....

THE DEPARTMENT OF NUCLEAR MEDICINE

I am hoping one of your readers might be able to offer us some advice on a problem which has been troubling us from some time.

We would like to arrange for background and foreground jobs to make simultaneous use of extended memory (ie. above 32K words) for array storage with no overlapping.

In our application (GAMMA-11) the typical foreground job is FGAMMA, supplied by DEC ready assembled. A new version just released (V3.Ø) permits FGAMMA to collect data using low extended memory, which considerably enhances GAMMA-11's foreground data collection capabilities.

Our problem is that several FORTRAN image processing programs which we have written to store arrays in extended memory (by means of the VIRTUAL statement) will no longer run simultaneously in the background.

The FORTRAN User's Guide p. 2-5 states that the XM monitor must be used if background and foreground jobs are to make simultaneous use of extended memory, but I am still in the dark as to whether the XM monitor will automatically control where in extended memory FORTRAN VIRTUAL arrays are placed. If there is a way of achieving our aim using the smaller FB monitor, this would of course be preferable.

I will be most grateful for any suggestions.

Yours sincerely,

Roger Fulton

Roger Fulton

Physicist

RT-11 SPOTLIGHT

The RT-11 Spotlight allows users the opportunity to describe any system they may be currently involved with. Ian Hammond has submitted a comprehensive description of the 'STAR-eleven' multicomputer system. Other articles describing this system will be in future issues of the 'minitasker'. User input is always welcome in the RT-11 Spotlight.

The STAR-eleven multi-computer system

Ian Hammond
HAMMOND-software, Am Feldborn 22
D-3400 Goettingen - West Germany

STAR-eleven is a general-purpose multi-computer system based on DEC's single-user system RT-11. A STAR-eleven system consists of a host PDP-11 running an RT-11/FB system, and upto eight satellite PDP-11 computers, each of which support a full, high quality RT-11/SJ environment.

- The host

The foreground of the host RT-11/FB system is used to run the satellite 'supervisor' which provides satellites with access to all systems peripherals, and enables intra-satellite communications.

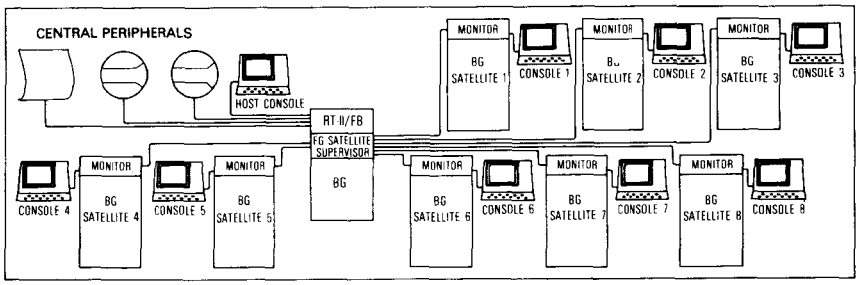
The host background is free, and may be used as an RT-11 workplace. In larger network configurations the host background is often used to run the STAR-eleven 'software cache' - the software cache accelerates satellite i/o thruput.

- The satellites

The crucial feature of STAR-eleven is that it provides each satellite with a full, fast RT-11 environment. The implication of full support of RT-11 is that all RT-11 programs run on satellites without modification - indeed, the same programs run on the host as on the satellites. Satellites have fast access to all host peripherals (disks, printers etc.).

A user approaches a satellite in same way as a standard RT-11 system, and no new procedures must be learned. All editors, compilers and utilities run on satellites. Indirect command files and a subset of the interactive console command language are supported.

The satellite monitor is a complete rewrite of RT-11 for the satellite environment and packs all RMON, USR and basic KMON functionality into 2.6k words, leaving work spaces of upto 27k words. Satellites are connected to the host using fast error-free communications devices specially developed for local networks. The combination of low monitor overheads, fast connections and host optimizations, produces a satellite work place that is often faster than a standard RT-11 workplace.



- Mini-computer systems

For many applications the modern single-user mini-computer is the ideal system. The user has complete control of the machine, and the machine can be located close to the work place. These characteristics are very important in the support of real-time experiments - local laboratory peripherals can be connected directly to the computer in the lab, and the computer guarantees fast interrupt response times.

However, in providing a number of users with computing facilities the single-user system is expensive. Each single-user system requires its own systems peripherals, and when magtapes, printers, plotters etc. are considered, the expense becomes prohibitive.

The traditional answer to providing a number of users with computing facilities is the 'multi-user' system. While the multi-user system does provide a number of users with shared access to processor and peripherals the very act of sharing these resources creates some problems. Firstly, because the processor is centralized, laboratory peripherals must be connected over long distances, and secondly, because processor time is shared between a number of users it becomes difficult to guarantee fast interrupt reponse times. These constraints, and the added complexity of the multi-user monitor, make multi-user systems more difficult to use than single-user systems.

The 'multi-computer' system is an alternative to the 'multi-user' system in providing a number of users with computing facilities - the difference is that the multi-computer system retains - and enhances - all the advantages of single-user computers. With a multi-computer system shared peripherals, disks and printers, are centralised on the host. However, local laboratory peripherals are connected directly to the satellite in the lab. Since each satellite is a 'single-user' machine it can guarantee fast interrupt response times. A satellite computer is even simpler than a stand-alone single-user computer, and is correspondingly easier to use.

Multi-computer systems are more reliable than a multi-user computer. The multi-user system centralises all peripherals on a single processor, and thus failure, maintenance or new development of any device causes downtime for all users on the system. With a multi-computer system there is a natural separation of devices between host and satellites. Host devices tend to be more reliable and require less maintenence than satellite devices. More importantly, failure of a satellite does not affect the system or other satellites. Thus, a satellite can be repaired, or new devices can be developed for a satellite without causing any system downtime at all.

Some STAR-eleven installations use 'portable satellites' to achieve very high reliability and flexibility. In this case a number of satellites are constructed with identical hardware configurations, and a 'computer plug' is installed in each laboratory. An experimentor simply rolls a satellite into the laboratory and plugs it in. If the satellite fails during an experiment another satellite can be used to replace it. This approach allows satellites to be shared between a number of laboratories, and to provide very high reliability.

- Decomposing the multi-user computer

In the previous section we showed that the multi-computer system was alternative to the multi-user system in providing multiple users with computing facilities. In this section we develop that theme in a semi-formal way by showing the relationships between single-user, multi-user and multi-computer systems. The point of this derivation is that it shows clearly the position of the multi-computer system in the computer landscape.

Single-user and multi-user systems both provide the same basic services: Firstly they provide a 'job space' - i.e. an environment in which programs can run.('Job-space' thus includes both the computers machine language - its hardware instruction set- and operating system language-its software instruction set, system calls, etc.) Secondly, using this 'job space' and additionally an 'interactive console command language' they provide the user with a working environment - which we call a'user space'.

Using these two terms ('job-space' and 'user-space') we can classify most systems as follows:

System type	Classification
single-user,single-job	one userspace, one jobspace
single-user,foreground/background	one userspace, two jobspaces
multi-user,multi-job	multiple userspaces and jobspaces

Table 1. System Classifications

The multi-user computer can be described as emulating a number of single-user computers and, since its emulating a number of computers, its emulating a network. If we decompose this 'virtual' network into a physical network, by providing each terminal with its own computer, then each of the terminal computers will provide a user with a jobspace and userspace.

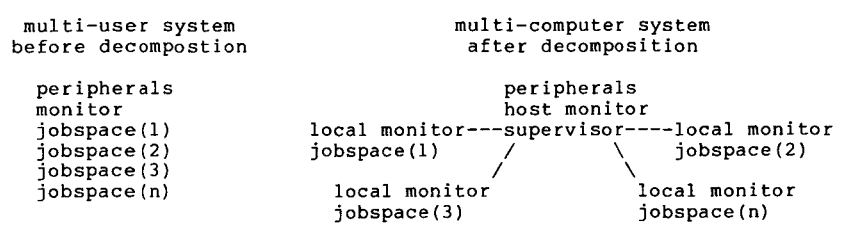


Figure 1. Decomposing the Multi-user computer

A question then arises: What remains of the central computer? Since we dont want to provide each terminal computer (satellite) with a separate set of peripherals, it is clear that the systems peripherals must remain on the central computer (host). Therefore, the hosts task is to provide terminal computers with access to the central systems peripherals.

One important point is, that the host is no longer a multi-user computer. Indeed, if the host remains a multi-user computer, we will have failed to decompose it! The host has the single task of providing satellites with access to common peripherals - and a single-user system is well-suited to that task. Since, each of the terminal computers is also a single-user computer, what we have derived is a multi-computer system, constructed entirely from single-user computers.

- Multi-computer extensions of RT-11

RT-11 is a 'single-user' system, and some extensions to RT-11 were required in expanding it to a multi-user environment. However, these extensions are user-transparent and compatible with existing stand-alone RT-11 systems.

STAR-eleven supports lineprinter spooling of all satellite and host line-printer output. The spooler keeps individual users output together and in order. The spooler is completely automatic and is user transparent.

Intra-satellite messages are implemented by using the RT-11/FB intra-job message requests .SDAT and .RCVD. Messages must have a header and length that conforms to the STAR-eleven standard. Messages are used to send messages, or console commands, to other satellites. These facilities are also available at the console command level with two new commands 'SEND' and 'TELL'. The TELL command will send a console command to another satellite. These facilities are used to initiate and synchronize parallel processing tasks on multiple satellites.

Messages can be sent to and from the host, the host program 'STAR' can send messages to one or more satellites. Command files can also be sent to the host for execution. The host job supervisor logs the output of such files and notifies satellites of job completion.

Some monitor functions need to be localised on the satellites. Each satellite has an individual start-up command file to handle satellite set up on bootstrap. Each satellite also has a local 'ASSIGN' facility which allows satellites to have separate logical device assignments.

In a given installation some satellites may possess different arithmetic hardware or console types. STAR-eleven has a user-transparent mechanism that will select the correct version of a program for the satellites hardware. For example, in response to the command 'R BASIC' this mechanism would run a FIS version of BASIC on FIS satellites, and an FPP version on FPP satellites.

The systems activity and status can be monitored on the host with the program 'STAR'. 'STAR' provides various reports of satellite status, files open, and so on. It also provides statistics of satellite requests and cache hits. On satellites the program 'NEWS' prints the daily news and reports general system status.

The program 'CREATE' is used to guide the systems manager thru network generation procedures. Network generation is relatively simple and can be accomplished in under an hour. Satellites can be reconfigured without rebootstrapping the system.

- Satellite performance

One main goal of STAR-eleven is to provide a satellite working place that is as fast as a stand-alone RT-11 workplace. A lot of factors affect satellite speed:

CPU SPEED - Raw host/satellite communications speeds are determined by the time required for the interrupt service routines. Thus, an 11/34 satellite will be faster than an LSI-11 satellite.

JOBSPACE - MACRO, LINK, FORTRAN and PIP all like the large satellite jobspaces of more than 25k words. The size of the satellite jobspace is unaffected by the number of handlers loaded on the system. Satellites with extended memory can use the FORTRAN VIRTUAL feature to get at extended memory.

SOFTWARE CACHE - Systems thruput is dramatically increased by the software cache. The cache operates on file structured reads from random access devices. If a requested block is not in the cache, then STAR-eleven reads 4 blocks into the cache. Since the cache has a hit rate of upto 75%, 3 out of 4 read requests can be satisfied from the cache without a disk access. The cache reduces disk activity when many users are active; the cache also enhances the performance of the large number of programs that use 'block at a time' i/o.

VIRTUAL I/O - The commands COPY and PRINT use a special form of I/O in which no data is physically transferred between host and satellite. This feature accelerates the performance of these commonly used commands.

USR BYPASSES - STAR-eleven overcomes certain USR bottlenecks that occur in a multi-user environment with the following enhancements: Magtape directory operations are performed outside the USR, DECtapes are positioned over the directory before calling the USR, and a .DSTATUS software cache satisfies 99% of all .DSTATUS requests outside the USR.

The satellite connection enables disk/satellite transfers of upto 10k words per second. Combining this raw speed with a unique software cache system on the host, and low satellite monitor overheads, STAR-eleven satellites often out-perform equivalent stand-alone RT-11 systems.

- Satellite program request support

The following table details the program request support available on satellites. Satellites also support overlays, 'exit to command file', and all relevant fixed monitor offsets. Satellites have full support for asynchronous i/o requests and completion routines.

TTYIN	TTYOUT	TTINR	TTOUTR	PRINT	GTLIN	RCTRLO
DATE	GTIM	GVAL	SFPA	TRPSET	GTJB	SCCA
SETTOP	HERR	SERR	INTEN	SYNCH	MFPS	MTPS
LOOKUP	ENTER	DELETE	RENAME	DSTATUS	CSISPC	CSIGEN
CLOSE	PURGE	WAIT	REOPEN	SAVESTATUS		
READ	READW	READC	WRITE	WRITW	WRITC	SPFUN
SRESET	HRESET	EXIT	CHAIN			
SDAT	SDATW	SDATC	RCVDC	MWAIT		

FETCH and RELEASE are ignored, handlers must be 'LOADed' on the host. LOCK and UNLOCK are ignored, satellites may not lock the USR.

- Summary

A multi-computer system provides a new alternative to the problem of providing a number of users with computer facilities. Single-user systems can grow into multi-computer systems without losing the advantages of a single-user environment, and without rewriting their software for a new operating system.

The multi-computer system provides a natural, user-transparent, method to take advantage of the low prices of processors and memories, by applying distributed processing to the 'operating system' itself. The multi-computer system separates central tasks and peripherals from local processing and peripherals to implement a well balanced system that has advantages in both reliability and flexibity.

STAR-eleven is perhaps the first system that can be truly described as a full 'general-purpose multi-computer system'. STAR-eleven has been refined over the past five years to exploit the inherent potential of distributed processing. Providing satellites with a full, fast operating system environment changes the name of the game for local networks and opens a new area of computing possibilities, especially for single-user computer users.

- Acknowledgements

The author would like to thank the department of Neurobiology of the Max Planck Institute in Goettingen which provided the environment for the growth of STAR-eleven. In particular Dieter Michael and Werner Buchholz must be thanked for their close collaboration. I would also like to thank Chris Ryan for his help with this paper.

PAST SYMPOSIUM INFORMATION

Starting with the Chicago 'Wish List' we will allow RT-11 SIG members to vote for the items that they feel are the most important to them. This ordered list will then be forwarded to the RT-11 development group for a formal response.

RT11 1979 San Diego Wish List

The following wishes were submitted to DEC at the San Diego DECUS Symposium in December. Some tentative answers are where indicated.

1. Add parameters to indirect command files. Example: comfil parm1 parm2; call a listing. See TSO as a possible model.

a. Difficult in RT11. Like the idea.

2. I would like 'multiline printer' support under the SJ Monitor. I have a MINC with a DLV-11J Serial Interface. It would be nice to be able to direct monitor or program (FORTRAN) output to any one of the three extra serial ports on the DLV-11J. I can send output to one of the ports with the Serial LP Handler supplied with the MINC/BASIC software. If the RT11 SJ monitor had LP0:, LP1:, and LP2: as device names corresponding to the three extra SLU ports, then I could direct output to any one of three serial devices that I attach to the SLU ports.

a. Good idea, but line printer would have to be immense and complicated.

3. SLP to work on both - either (a) distribute commented sources as standard (instead of, not in addition to), or (b) make necessary changes to comment stripper so that SLP will work the same on commented and uncommented.

a. No. We are going to try to get patches from commented sources. Automatic distribution floppies perhaps.

4. RT11 assembly, LIBR and LINK support on VAX/VMS and RSX11M. This would allow software development on the VAX or large 11 and then down loaded to a smaller 11 running RT11. Emulation of all of RT11 is not required as the applications are laboratory control and data acquisition.

a. Version 4 documents this. There is a crash dump for RT11. You can get it from your local resident specialist.

5. Provide a means for the user program to determine if input via .GTLIN comes from the terminal or from an indirect commands file. This would allow programs to handle errors differently if the input comes from a file.

a. Version 4 documents this.

6. COPY/dev copies bad block areas from source disc - also does not recognize bad blocks initialized on object disc. If it did, we would use it.

a. Use initializing/bad or/replace and then use a squeeze.

7. DIR has some nice switches like since, before and alpha. Can there be an add

a. Will consider it.

8. Would like FORTRAN OTS Internals Manual.

a. No plans for manual.

9. RT11 V03B Software Support Manual as described for V4. Those of us with products already using RT11 V03B may not be able to justify a change to V4. support manual is highly desireable. Note: V4 looks very promising.

10. FORTRAN OTS Internals Manual comparable to the old FPM11 Manual.

11. FORTRAN - stop changing and/or resequencing the patches... If you must correct a patch, issue a new one to correct. it costs far too much to back and start over.

a. We won't change anymore..

12. FORTRAN - how about ANSI compatible access to elements of virtual arrays, i.e., CALL SUBR(A(I)) rather than CALL SUBR(A,I).

a. May consider.

13. Would like optional subscript bound checking on FORTRAN 4.

a. No.

14. Would like to be able to copy RT11 files (.MAC, .OBJ) to XXDP format.

15. Quite often we find the preformatted floppys are corrupted and sometimes become corrupted after use. It would be nice to be able to reformat them without having to bring down RT11.

16. To increase throughput on my floppy-based system, my startup command file runs a tiny program which patches the monitor (in memory) to disable program swap-out at program abort or .EXIT. (I thus readily give up the ability to restart. The patch just changes a conditional branch to an unconditional branch.) It took a long time to find the proper patch working with the distributed, uncommented source code; I don't look forward to repeating this for V4. Thus I wish for a NOSWAPOUT (or NORESTART) command; or,failing that, documentation on how to make the patch. V4 looks really good. I appreciate your work.

17. I would like 2 word blocks in large disk support.

18. I would like a file of SET options to have 'sticky' effects, such as SET LP LC. would like (1) a SHOW interface for SET options in a device driver (including TT function done by monitor). Also (2) to be able to SHOW other things that are set, i.e., SHOW USR without having to get all of SHOW CONFIG output.

19.1. 'Failsafe' copy to move all of a large device to multiple small ones (such as OS/8 FOTP '/F') - backup of named files from RK/RL/etc. to RX02.

19.2. New logical device name like CM to use as defaul T for indirect file input. would allow DK to be assigned more easily - dual floppy as system device.

19.3. Option to always move a file when use copy to same device.

19.4. Additional date options for PIP, as in DIR, to allow COPY, RENAME, etc.

20. Publish or distribute RT patches in source form for source licensed customers.

21. Desire ability to put SWAP.SYS on device other than SY. Reason: We have an application that requires many file opens and closes at regular intervals. We are tight for memory so we cannot SET USR NOSWAP. we have a solid state disk which would be ideal for swapping. (it is too small to hold the monitor and swap sy and application related data).

22. I know some of these are coming in V4, but this list was printed before San Diego DECUS: print the physical device and unit number at beginning of all directory listings. How about a /BEGIN switch for the COPY command? It would be nice if file creation times were kept, as well as dates. This would, among other things, allow DEC-10-like compil-class commands to be implemented, which would execute an entire compile,link/run sequence while only recompiling those source files with creation/date/time later than the corresponding .OBJ files.

You should be able to type in text to a variable in batch. When you power on the printer (LA-180), and a program is trying to write to it at that time, one or more characters are usually lost. Implement image mode input from the terminal in image mode, allowing any and all control characters to pass, even those with a predefined meaning to RT11 (like S and Q). Please bring back the capability to easily (i.e., in one step) move a file down into an empty slot in the directory. You formerly could do this by copying the file to itself with the /X option to PIP. I do this all the time to fill empty slots without having to squish the disk--it can be much faster (and safer). The trouble is, with 600 files on the disk, a COPY followed by a RENAME takes a long time too. A switch on the COPY command to force a physical copy even when source and destination filenames are the same would be appropriate.

BATCH doesn't always work properly under V38. Sometimes a program-detected error (e.g., 'file not found' by the linker) will cause BATCH to hang up with no error message. Implement a verify option on copy operations, especially on SQUEEZE and COPY/DEVICE read each block after it's written to verify we can get it back again. The TYPE and PRINT commands shouldn't assume an extension of .LST. Other copying commands don't. At least, let them try a null extension if .LST isn't found.

On loss of copied, renamed, and deleted files, use the physical device and unit number. Allow an indirect command file to chain to itself (or another) indefinitely without nesting. SET TT CONSLOL=0 given from terminal 0, causes system to hang. Allow a /NOBOOT switch on the SQUEEZE command to suppress automatic booting when the system device is squeezed. Note: at DECUS you said this was left out of the

command language on purpose. I still see no reason not to allow this to be done as a one-step operation. It so happens I do it fairly frequently.

Allow a remote terminal to be the console device, so you can support remote installations while staying at your home base.

Allow a foreground job to chain to another foreground job, assuming of course enough memory is available. Implement SYSGEN options for useful monitor calls, SET SET TT CRLR, etc, and so on that aren't normally in the SJ monitor, so we can put them in if we want. When the /B switch is used with the linker, have an option to suppress the garbage disk block written out at the front of the .OBJ file. This for TSD with B:100000 and then run a utility to zap the 63 unwanted blocks. If you're doing a lot of links at once (a not infrequent occurrence wherever continuing development is going on), or if your system happens to have only floppies, you can easily run out of space. Even though you can get around it, it would be so much cleaner (and faster) for those extra blocks never to appear at all

Have the linker print the high address of the linked program on the terminal, whether or not a map was asked for. Be sure this is done so that the information will appear on the log file if the link is run under BATCH (a) document the maximum number of logical assignments, (b) make this number independent of the number of device slots.. I want a dozen or so logical assignments with minimum memory penalty.

After deleting files, tell total number of blocks freed. The following KMON commands differ in only one detail, yet the first one CSI-F-Illegal command':

```
  COP DK:(111111,222222,333333,444444,555555).REL DK2:
  COP DK1:(111111,222222,333333,444444,555555).REL DK2:
```

the following commands, however, will work:

```
  COP DK1:(111111,222222,333333,444444).REL DK2:
  COP DK1:(1,2,3,4,5).REL DK2:
```

Allow DEL, H, and U to be end-of-line terminators. This will facilitate using line-mode input instead of character mode. Implement a /SUBSTITUTE switch to LINK to allow linking in a subroutine with a different entry point than the calling routine calls for. People with a lot of programs (line business users) can use this to save disk space and directory space by having one main program file which calls (at least) two external routines, 'XXXXX and YYYYY', which share the same overlay region. XXXXX does initialization and YYYYY does processing. At link time, an indirect command file specifies which global symbols to resolve to XXXXX and YYYYY. In this way, I might be able to use only a few main program files for nearly 200 separate programs. That's a lot of space saved.

Modify the LP handler to suppress trailing blanks--a real advantage on character printers like the LA-180. You may want this controlled by a SET command. The command .DELETE X,Y should assume null extensions on X and , and not interpret it as X.* and Y.*. Besides being very dangerous, this is inconsistent; the command .DELETE .X,.Y does not assume *.X and *.Y (in fact, it gives an error).

PIP should diagnose and give an error for the case of a command to COPY/PREDELETE to the SAME device. such a command will never be

given except in error, and the result is the loss of all the files you thought you were copying!

Marilyn L. Runyon
RT11 Wish List Coordinator

RT-11 MARKETING

B I C A R - A BILLING, INVENTORY CONTROL AND ACCOUNTS RECEIVABLE SYSTEM FOR THE PDP-11

BICAR is a generalized software package developed to perform the functions of billing, inventory control, accounts receivable and sales analysis on a small PDP-11 system. It is highly operator oriented with a large number of operator controllable options and extensive data entry verification checking. The package has been configured to run under the RT-11 operating system and is designed specifically for the small business entity. Written in FORTRAN IV, it requires RT-11 3B or later.

For further information contact:

Software Dynamics Inc.
1000 Yale Avenue
P.O. Box 500
Wallingford, CT 06492
(203) 269-4998

TAPE COPY

7 Apr 80

****
* TAPE COPY ERROR *
* NOTICE *
****

I inadvertently implied in the February 1980 'Mini-Tasker' that SIG tape copies were available from the DECUS library for $20. The following paragraph states the situation correctly.

Starting with the Spring 80 DECUS symposium we will join with several of the other SIG's and let the DECUS U.S. Chapter handle all reproduction and distribution of our tape. We will continue to accept submissions (on floppy disk and mag tape) and build our master tape at the symposium. LUGs will receive a free copy of the tape to distribute as they see fit. Others may order copies at the symposia for a small fee. After some period of time copies of the tape will be available from the DECUS U.S. Chapter for their then current (and higher) fee.

N. A. Bourgeois, Jr. / 1738
Systems Engineering Division
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185

DECUS

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

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.