THE mini-tasker

DECUS
RT-11 SIG NEWSLETTER

SEPTEMBER 1980 VOL. 5 NO. 4

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

I would like to remind all LUG's that the "minitasker" is available to disseminate information among your respective members.

USER INPUT

I was particularly interested in the article in the June 1980 mini-tasker on problems with the Fortran IV compiler using inline code. I experienced similar problems while installing some canned plot software. The statement "I3=3" caused an illegal memory reference when executed. I was able to eliminate this problem (through trial and error) by disabling global register binding during compilation.

I have also had the compiler fail twice in compiler routine "Regalo". This problem occurred on a statement that appeared elsewhere in the program in identical form. The problem disappeared if I removed the problem statement. This was not a satisfactory solution however, so I made the statement into a subroutine and replaced both occurences with calls to the subroutine. I hope these solutions may be of some assistance to other users.

In addition, I am interested in communicating with other RT-11 or RSX-11M users involved in space physics research for the purpose of exchanging software and/or ideas. My address is:

Steve Griffin
Space Physics Group
University of Maryland
Department of Physics
College Park, MD 20742
301-454-6444
TWX #710-826-1125.

Sincerely,

Stephen P. Griffin

Stephen P. Griffin

Terence I Hale
Nuklearmedizin
Kantonsspital Schaffhausen
Schaffhausen
CH-8200 SWITZERLAND

Dear Mr. Demens,

As a small nuclear medical department we have problems in using our indispensable computer to its full (or half full) capacity. To pay the bills and to keep the bailiff away we have developed a computer concept for a small nuclear medical department, whereby free computer time is efficiently used for other applications in the hospital.
Within this concept we have developed a number of data bank applications to deal with hospital stock and inventry control, drug stock and inventry control of our pharmacy department, bacteria resistence to antibiotics, patient bed dispossesion in the hospital.
Any body interested in the programs may obtain futher information from me.

Regards

Terence I Hale

Runs under RT-11

Dear Mr. Demers,

I send you a new program for management of files with application to the gestion of instructions for use programs in a computer system.

I thank you for publication.

Sincerely yours

Daniel GUINIER

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

APPLICATION NOTE :   RT 11 (FORTRAN IV SOURCES.)
****

MANAGEMENT OF CHARACTERS FILES BY AN INDEXED-RANDOM ACCES METHOD WITH VARIABLE LENGHT RECORDS.

APPLICATION TO THE GESTION OF INSTRUCTIONS FOR USE PROGRAMS IN A COMPUTER SYSTEM.

****

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   FRANCE

INTRODUCTION :
****

THE CONVIAVILITY OF A COMPUTER SYSTEM CAN BE INCREASED IF THE INSTRUCTIONS FOR USE IT ARE INCLUDED ON THE SITE OF THE OPERATING SYSTEM.

OUR OBJECT WAS TO STUDY AND REALISE A PERFORMING METHOD TO ENSURE ITS MANAGEMENT AND TO PROVE ITS VELOCITY; WE HAVE CHOSEN A RANDOM ACCESS STRUCTURE WITH AN INDEXED FILE, THE RECORDS OF THE CHARACTERS FILE HAVING DIFFERENT LENGHTS.

PRINCIPLE
****

1). THE CHARACTERS FILE (MODE.EMP) IS CREATED UNDER THE CONTROL OF THE TEXT EDITOR EDIT. IT CAN BE MODIFIED AS REQUIRED. THE STRUCTURE OF EACH LOGICAL RECORD BEGINS BY THE "$" CHARACTER AND THE SIX CHARACTERS OF THE NAME OF THE PROGRAM. (EX. $FIRST ,$SECOND,$ONE   ,..), IT CONTINU WITH THE CONTAIN OF THE DIRECTIONS FOR USE IT. THE END OF THE CHARACTERS FILE IS MARKED BY THE "@" CHARACTER.

2). THE INDEXED-RANDOM ACCESS FILE (MODE.IND) IS CREATED AND SORTED IN INCREASING ORDER (D.GUINIER AND R.KIRSCH (1980)) BY THE FIRST PROGRAM "MODE", AFTER THE CHARACTER FILE (MODE.EMP) HAS BEEN REDACTED.

THE STRUCTURE OF A RECORD OF THE INDEXED FILE IS ONE REAL*8 (FOUR WORDS). THIS STRUCTURE REPRESENTS THE TRANSFORMATION RADIX50 OF AN ENCODED CHAIN OF 12 CHARACTERS : SIX CHARACTERS ACCORDING TO THE NAME OF THE PROGRAM, SIX FOR THE NO. OF THE BLOCK AND THE POSITION IN THIS BLOCK AFTER INTEGER-ASCII TRANSFORMATION. THIS PROCESS IS EXECUTED FOR EACH PROGRAM.

3). THE PROGRAM "EMPLOI", AFTER DECODING THE NAME OF EACH PROGRAM YOU WANT THE DESCRIPTION FOR USE, SEARCHS THE NO. OF THE BLOCK AND THE POSITION IN IT WITHIN THE INDEXED RANDOM ACCESS FILE (MODE.IND) BY A DICHOTOMIC METHOD. THIS TECHNIQUE INCREASES THE VELOCITY. THE CONTENT OF THE LOGICAL RECORD IS PRINTED ON THE TERMINAL TT:

REFERENCE
****

D.GUINIER AND R.KIRSCH (1980)   COMPARISON OF SEVERAL SORTING METHODS FOR NUMBERS OR CHARACTERS RUNNING ON A PDP11/05 UNDER RT11 IN FORTRAN IV LANGUAGE DECUS MINI-TASKER VOL.6 NO.2 APRIL 1980.

PROOF :
****

A COMPARISON HAS BEEN PERFORMED WITH "MODE.EMP" CORRESPONDING TO 36 PROGRAMS (66 BLOCS OF 256 WORDS), THE ANSWER TIME WAS LESS THAN ONE SECOND WITH THE METHOD DESCRIBED AND 18 SECONDS WITH A SEQUENTIEL METHOD.

REMARK :
****

A NEGATIVE DECODING REAL*8 NUMBER CAN OCCUR, DUE TO THE IRAD50 MODULE OF THE SYSLIB LIBRARY; HOWEVER THIS HAS NO EFFECT ON THE SEARCH, ONLY ON THE SORT.

LISTINGS : PROGRAMS "MODE" AND "EMPLOI"
****

1) LISTING OF "MODE" :

```
FORTRAN IV      V01C-03A

0001        BYTE BUF(512),NOM(12),NCODE(6),SIGN,FIN
0002        REAL*8 A(2000),AMAX
0003        DATA NENR,NMOTS/512,256/INDX1,INDX2/2*1/
           1SIGN,FIN/'$','@'/MAX/0/
      C
0004        CALL ASSIGN(1,'MODE.EMP',0,'OLD')
0005        CALL ASSIGN(2,'MODE.IND',0,'NEW')
0006        DEFINE FILE 1(NENR,NMOTS,U,INDX1)
0007        DEFINE FILE 2(NENR,     4,U,INDX2)
      C
0008        NCAR=NMOTS*2
0009        DO 1 NBLOC=1,NENR
0010        READ(1'INDX1)BUF
      C  SEARCH "$" AND "@"
0011        DO 2 NP=1,NCAR
0012        IF(BUF(NP).EQ.FIN)GO TO 5
0014        IF(BUF(NP).NE.SIGN)GO TO 2
      C  NOM(6) GENERATION
0016        DO 3 K=1,6
0017  3     NOM(K)=BUF(NP+K)
      C  ENCODE ASCII : NBLOC AND NP
0018        ENCODE(6,100,NCODE)NBLOC,NP
0019  100   FORMAT(2I3)
      C  NCODE IN NOM
0020        DO 4 K=1,6
0021  4     NOM(K+6)=NCODE(K)
      C  TRANSFORMATION OF NOM : ASCII-RAD50-REAL*8  (NOM-A)
      C
      C  REAL*8 CAN BE NEGATIVE NUMBERS.
           C
  0022          MAX=MAX+1
  0023          CALL IRAD50(12,NOM,A(MAX))
  0024   2      CONTINUE
  0025   1      CONTINUE
         C  A IS SORTED
  0026   5      CALL SORT(A,1,MAX,1)
         C  CREATING THE INDEX FILE "MODE.IND"
         C  1) NUMBER OF ITEMS
  0027          AMAX=MAX
  0028          WRITE(2'INDX2)AMAX
         C  2) ITEMS THEMSELVES.
  0029          DO 6 I=1,MAX
  0030   6      WRITE(2'INDX2)A(I)
  0031          STOP
  0032          END
```

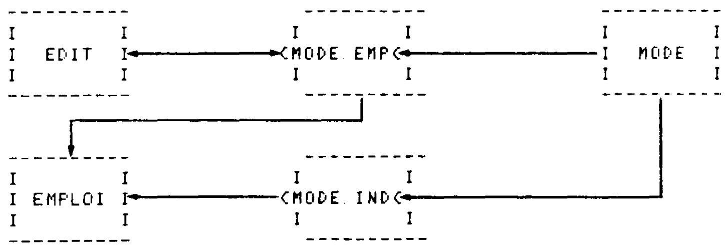
2) LISTING OF "EMPLOI" :

```
FORTRAN IV      V01C-03A

0001        BYTE BUF(512),NOM(6),ART(12),SIGN,FIN,BLANC
0002        REAL*8 A
0003        DATA NENR,NMOTS/512,256/INDX1,INDX2/2*1/L/0/
           1SIGN,FIN,BLANC/'$','@',' '/
```

```
0004        DATA LEC,IMP/5,7/
      C
0005        CALL ASSIGN(1,'MODE.EMP',0,'OLD')
0006        CALL ASSIGN(2,'MODE.IND',0,'OLD')
0007        DEFINE FILE 1(NENR,NMOTS,U,INDX1)
0008        DEFINE FILE 2(NENR,     4,U,INDX2)
      C
0009        NCAR=NMOTS*2
      C  NUMBERS OF ITEMS IN "MODE.IND"
0010        READ(2'INDX2)A
0011        MAX=A
      C
      C  CREATING THE ITEM TO SEARCH
0012  9     WRITE(IMP,200)
0013  200   FORMAT(//'$DIRECTION FOR USE THE PROGRAM ')
0014        READ(LEC,300)NOM
0015  300   FORMAT(6A1)
0016        IF(NOM(1).EQ.BLANC)STOP
0018        CALL IRAD50(6,NOM,Y)
      C  DICHOTOMIC SEARCH IN THE FILE "MODE.IND"
0019        NINF=0
0020        NSUP=MAX+1
0021  1     K=(NSUP+NINF)/2
0022        READ(2'K+1)A
0023        CALL R50ASC(12,A,ART)
0024        DO 2 J=1,6
0025  2     NOM(J)=ART(J)
0026        CALL IRAD50(6,NOM,X)
0027        IF(Y.EQ.X)GO TO 3
0029        IF(Y.LT.X)NSUP=K
0031        IF(Y.GT.X)NINF=K
0033        IF(NSUP-NINF.NE.1)GO TO 1
0035        STOP ' NAME NOT FOUND !'
      C  NO. OF THE BLOCK AND THE POSITION
0036  3     READ(2'K+1)A
0037        CALL R50ASC(12,A,ART)
0038        DECODE(12,400,ART)NOM,NBLOC,NP
0039  400   FORMAT(6A1,2I3)
0040        NP=NP+1
0041        INDX1=NBLOC
0042  4     READ(1'INDX1)BUF
0043        DO 5 I=NP,NCAR
0044  5     IF(BUF(I).EQ.SIGN.OR.BUF(I).EQ.FIN)GO TO 6
0046        I=NCAR
      C  PRINTING ON THE TERMINAL TT:
0047  6     DO 7 J=NP,I
0048  7     K=ITTOUR(BUF(J))
0049        NP=1
0050        IF(BUF(I).NE.FIN.AND.BUF(I).NE.SIGN)GO TO 4
0052        DO 8 J=1,71
```

```
FORTRAN IV        V01C-03A

      8       K=ITTOUR(BUF(I))
              K=ITTOUR("12)
              K=ITTOUR("15)
              GO TO 9
              STOP
              END
```

SYNOPTIC
****



BUTLER, NJ - TSX, the time sharing system used with Digital Equipment Corporation's RT-11 operating system, now has an independent users group. Membership is open to all licensed TSX users, as well as users of DPS word processing software and the well known RTSORT utility.

The TSX Users group was formed indpendently of, but with "blessings" from, S&H Computer Systems, Nashville, the developers of TSX. The primary purposes of the group are user-to-user communications and mutual assistance, user input for new software developments, and an exchange library. A hardware site listing and newsletter are planned, as is an annual meeting.

An application can be obtained by writing the TSX Users Group, Box 218, Butler, New Jersey 07405.

"WILDCARD" FILE GROUPS AS INPUT TO FORTRAN PROGRAMS

Alfred Kracher

Naturhistorisches Museum, Mineralogisch Petrographische Abteilung
Burgring 7, A-1014 Wien, Austria

In our application, we usually apply a number of different (FORTRAN) data reduction programs to groups of files with similar attributes (for example, list all files with extension .DAT on the printer). In other words, we need to simulate "Wildcard" input to FORTRAN programs. The method I use to achieve this, and which is described here, may not be the most elegant solution, but it's very flexible and straightforward even to inexperienced programmers. In the following example, it is assumed that we want to apply a FORTRAN program whose name may be supplied by the user ("user program") to all files whose extension (file type) starts with digit "2" (in our nomenclature, this happens to be one type of data file). To do this

(1) type @ DE

(2) DE. COM invokes DIR, and creates a directory listing file (SY: FILES. LST) of all files subsequently required. This is the "true" wildcard step.

(3) DE. COM invokes CRDCO, which will prompt the user for the name of the program (the "user program") to be applied to the files contained in SY: FILES. LST. This is the only operator intervention required in the process; users may find it convenient to let CRDCO supply the name, too, if there's only one "user program".

(4) CRDCO then creates a command file (SY: FILES. COM) which begins with

R "user program",

contains the names of all files from SY: FILES. LST, and ends with <CTRL> C to return control to DE. COM.

(5) DE. COM transfers control to FILES. COM, which executes the user program. After FILES. COM returns control to DE. COM, the temporary FILES.* are automatically deleted.

The user program has to contain the sequence

```
10    CALL ASSIGN (INUNIT,, -1)
      :
      :
      CALL CLOSE (INUNIT)
      :
      :
      GOTO 10
```

and it must not, of course, contain any READ's from the console.

Other users may think of their own variations; for example, let CRDO insert names of output files into FILES. COM, which respond to appropriate OPEN's in the user program(s). The following listing is just to stimulate your imagination.

```
         DE.COM:
!       Generate WILDCARD input for a FORTRAN program.
!
R DIR!                     Generate directory listing file
SY:FILES.LST=DK:*.2*/F/C:1!     containing the names of
!                          all required files.
R CRDCO!                   Convert directory listing to command file.
@SY:FILES!                 Call the command file written by CRDCO to
!       execute the program (whose name was supplied to CRDCO from
!       the console) for all files contained in FILES.LST.
DEL/NOQ SY:FILES!          Cleanup unit SY:
```

```
   0001          PROGRAM CRDCO
C ConveRt Directory listing into COmmand file
   0002          IMPLICIT BYTE (B)
   0003          DIMENSION B(13),B1(13),BH(15)
   0004          DATA BH/'R',14*' '/,NOUT/'^C'/
   0005          CALL ASSIGN(1,'SY:FILES.LST')
   0006          CALL ASSIGN(2,'SY:FILES.COM')
C Prompt for and read name of program to be executed by FILES.COM.
   0007  5       WRITE (7,1700)
   0008          READ (5,1110)L,(BH(I),I=3,MINO(10,L+2))
   0009          L=MINO(L+2,10)
   0010          LIN=1
C Count the number of lines in FILES.LST; the number of filenames is LIN-3.
C If the second line is empty, FILES.LST doesn't contain filenames. In that
C case, write an error message on TT:, and just <CTRL>C into SY:FILES.COM.
   0011  10      READ (1,1100,END=20)N,(B(I),I=1,N)
   0012          IF (LIN.EQ.2.AND.N.EQ.0) GOTO 100
   0014          LIN=LIN+1
   0015          GOTO 10
   0016  20      REWIND 1
   0017          LIN=LIN-3
   0018          READ (1,1100)N,(B(I),I=1,N)
   0019          WRITE(2,2200)(BH(I),I=1,L)
   0020          DO 30, II=1,LIN
   0021          READ (1,1100)N,(B(I),I=1,N)
   0022          NZ=0
C Throw all blanks out of the filenames.
   0023          DO 40, J=1,N
   0024          IF (B(J).EQ.' ') GOTO 40
   0026          NZ=NZ+1
   0027          B1(NZ)=B(J)
   0028  40      CONTINUE
   0029  30      WRITE (2,1100)NZ,(B1(I),I=1,NZ)
   0030          GOTO 50
   0031  100     WRITE (7,2710)
   0032  50      WRITE (2,2500)NOUT
   0033          CALL CLOSE(1)
   0034          CALL CLOSE(2)
   0035          CALL EXIT
```

```
C Formats
   0036  1100   FORMAT(Q,13A1)
   0037  1110   FORMAT(Q,8A1)
   0038  1700   FORMAT(' Progr.Name : ',$)
   0039  2200   FORMAT(10A1)
   0040  2500   FORMAT(A2)
   0041  2710   FORMAT(/' No appropriate files present on DK:'/)
   0042         END
```

DEC INPUT

The following was submitted by a DEC software specialist:

RT-11 V4
DOCUMENTATION CORRECTIONS AND ADDITIONS

The following is an unofficial set of corrections and additions to RT-11 V4 documentation; it is meant to complement the documentation articles published in the July 1980 RT-11 Software Dispatch. The information presented is correct to the best of my knowledge.

RT-11 SYSTEM RELEASE NOTES

CHAPTER 1 -- NEW FEATURES

On page 1-2, in section "1.2.2 TS11 Magtape", in the last sentence, change the number of supported TS11 controllers from eight to four. (This change should be reflected throughout all RT-11 software and documentation; this is a hardware, not a software, restriction.)

On page 1-3, at the end of section "1.3.2 System Jobs", add the following sentence: "Digital does not support user-written system jobs".

On page 1-7, add the following new section:

> 1.3.12 Distribution and Minimum Memory Requirements Changes
>
> The minimum memory requirements for the RT-11 V4 SJ and BL monitors has been raised from 8KW to 12KW. For this reason, RT-11 V4 is termed an upgrade from RT-11 V3B, rather than an update, so there is a separate QJ number (QJ012) for upgrading from RT-11 V3B. RT-11 is now distributed on RL02, but there is no longer any binary distribution of RT-11 V4 on DECtape or RK06, and no source distribution on RX02.

CHAPTER 2 -- CHANGES AND ADDITIONS TO EXISTING COMPONENTS

On page 2-2, in the second sentence of the fourth paragraph which begins "Code was added...", change the phrase "in extended memory" to "in the extended memory monitor".

On the bottom of page 2-7, add the following new paragraph: "SET options can now have octal values via a new bit defined in one of the the SET options table words."

On page 2-8, at the end of section "2.5 DUP", add the following sentence: "DUP now rereads detected bad blocks, thus, based on aural or visual observation of disk activity, DUP will appear to detect twice as many bad blocks as it reports."

On the bottom of page 2-8, in section "2.8 LIBR", add the following new option: "/F Supports FMS-11 forms libraries. This option is not described in the RT-11

documentation set, but in the FMS-11/RT-11 documentation."

On page 2-9, in section "2.9 LINK", add the following sentence to the end of the first paragraph: "Modules from non-multiple definition libraries referenced in only one segment will now be incorporated into that segment; only modules referenced in more than one segment or specified with the /INCLUDE (/I) option will be incorporated into the root."

On page 2-10, in section "2.13 ODT", add the following: "VDT can also be used for general RT-11 program debugging where locking out of other system activity by resetting of the CPU hardware priority to 7 (as ODT does by default) is unnecessary or undesirable. In addition, VDT's use of RT-11 terminal I/O ensures that any terminal output buffered by the user's program appears in its entirety before the breakpoint message indicating entry into VDT appears (unlike ODT's sometimes confusing operation). See section 4.9 on page 4-69 of the RT-11 Software Support Manual for additional information on VDT."

On page 2-12, add the following new section:

2.21 FORMAT

FORMAT can now format RK06 and RK07 volumes, and perform pattern verification of all block-replaceable volumes.

CHAPTER 3 -- CORRECTED PROBLEMS AND CURRENT RESTRICTIONS

On page 3-2, at the end of section "3.1.3 KMON", add the following sentence: "If any program created with the SAVE command under any RT-11 monitor (including earlier RT-11 versions) is run under the RT-11 V4 XM monitor without the above patch, the message "?KMON-F-Input error" results."

On page 3-3, add the following new restriction to the end of section "3.1.6 DUP":

4. Although the default device for the BOOT command is SY:, the default device for the monitor file name specified in the COPY/BOOT command is DK:. Although the RT-11 System User's Guide specifies that the input and output devices for the COPY/BOOT command must be the same, DUP does not check this. Therefore, check this carefully yourself when using the COPY/BOOT command.

On page 3-3, in section "3.1.7 LINK", delete the first paragraph; it is not a linker restriction. It implies that the user must check the link map when linking a background job with extended memory overlays; in reality, it is describing a condition which is automatically detected by

KMON when running a background program.

On page 3-5, in section "3.1.15 FORTRAN IV/RT-11", in the fifth paragraph, change the phrase "Running RT-11 In Less Memory Than Is Available" to "Setting Upper Limit on Memory Size".

On page 3-6, add the following new section to the end of section 3.1:

3.1.16 FILEX

FILEX does not support the KMON /WAIT option. If you must remove the system volume to perform a FILEX operation, you must run FILEX under the FB or XM monitor.

Digital format double density diskettes are not valid interchange format media. FILEX does not issue an error message if you mount a double density diskette in an RX02 drive to create an interchange format diskette; it appears to perform any operation requested. but actually does not. Therefore, be sure to mount only single density diskettes for use as interchange format media; the RT-11 format media can be in either density. (The density of a scratch diskette can be determined by initializing it first in RT-11 format. If a subsequent DIRECTORY command reveals 486 free blocks, the diskette is single density formatted.)

(The above restrictions also apply to the version of FILEX distributed with RT-11 V3B.)

CHAPTER 4 -- PROCEDURES AND PATCHES

On page 4-3, in section 4.1.4, in the second sentence of the third paragraph, following the word "options", add the phrase "other than those specified above".

On page 4-5, add the following new section to the end of section 4.1:

4.1.10 SYSGEN

The requirement of 2000 free blocks to perform an "automatic" SYSGEN stated in the RT-11 Installation and System Generation Guide is an upper limit that may not be required if only one or two monitors and a typical number of handlers are to be generated. For example, a typical RK05 SYSGEN can be performed on a copy of the RT-11 RK05 binary distribution, with that single RK05 being the source input device and the binary and map output device.

Users with RX02 based systems do not have to perform the manual procedure described in Chapter 10 of the RT-11 Installation and System Generation Guide in order to do a SYSGEN; the general procedures described in Chapters 8 and 9 can be used, as long as only one, or at most two, monitors are generated during each SYSGEN process. Begin the procedure by creating a working system disk as described in section 10.1 by using the patched distribution copies of the RT-11 V04 binary RX02 1/4 and 2/4. In addition to the files listed in section 10.1, also copy MACRO.SAV, SYSMAC.SML, and LINK.SAV to the working system disk; the resulting disk should have no bad blocks. Next, run the SYSGEN program as described in section 10.2, and specify DY1: as the source input device (question 152) and DY0: as the binary and map output device (questions 153 and 154). (Generation of two monitors during a single RX02 SYSGEN has not been tested for all cases, but appears to work as long as DY0: is not used as the map output device.) Answer Y to question "155. Do you want to retain the system OBJs?". When the SYSGEN program completes, delete the files SYSGEN.SAV, SYSGEN.CND, and SYSTBL.CND from the system disk, and then squeeze it. Mount the patched distribution copy of the RT-11 V04 binary RX02 3/4 disk in DY1:. Type or print the file DEVBLD.COM, and ensure that all the device handler source files required are present on DY1:. Any source files that are not present may be copied to DY1: from the patched distribution copy of the RT-11 V04 binary RX02 4/4 disk mounted in DY0: by using the COPY/WAIT command. Finally, copy the files SYCND.MAC and SYSTBL.MAC from DY0: to DY1: and type "@SYSBLD" to generate the new system. When this procedure completes, refer to the end of section 9.4.1 for final instructions.

In responding to SYSGEN question "25. Do you want floating point support [Y/N] (N)?", note that floating point support is required for FIS as well as FPU hardware. If you are unsure whether arithmetic hardware is present on the current configuration, issue the SHOW CONFIGURATION command.

SYSGEN question "89. Do you want line printer support [Y/N] (Y)?" really means "Do you want parallel (LP11/LPV11) support?". If you answered yes to this question, but really wanted serial line printer support, just hit <RETURN> to question "90. Does your line pinter have a non-standard vector or CSR address [Y/N] (N)?", and answer yes to question "93. Do you want serial line printer support [Y/N] (N)?". When DEVBLD completes, just ignore LP.SYG and rename LS.SYG to LP.SYS.

In responding to SYSGEN question "142. What is the CSR address of the first DZ multiplexor [O] (160010)?", note that the CSR address of the first DZ multiplexor is normally 160100, which is not the same as the default address supplied by SYSGEN.

Upon completion of a SYSGEN of a multiterminal monitor, a SHOW TERMINALS command issued under the newly generated monitor will verify that the information supplied (or defaulted) during the SYSGEN dialogue regarding interface CSR and vector addresses was correct. Missing logical unit numbers indicate incorrect CSR addresses; if a system crash occurs, a vector address was incorrectly specified. If a CSR or vector address has been incorrectly specified, and you have answered yes to SYSGEN question "155. Do you want to retain the system OBJs?", the entire SYSGEN process need not be repeated. Simply edit the file SYCND.MAC to indicate the correct CSR/vector address, and then reassemble the file SYSTBL.MAC and relink the monitor by manually issuing the corresponding MACRO and LINK commands as listed in the MONBLD.COM file.

Because MACRO has been made faster, a V4 SYSGEN takes approximately half the time as does a V3B SYSGEN; for example, a SYSGEN of a V4 multiterminal SJ monitor on a 30KW LSI-11 system takes approximately 35 minutes. MACRO remains primarily compute bound however, so that the time to do a SYSGEN is relatively independent of the source input and binary output devices (the manual RX01 SYSGEN, of course, being an exception).

RT-11 INSTALLATION AND SYSGEN GUIDE

The distributed DL, MT, MTHD, MM, amd MMHD handlers each support only two device units; the distributed MS and MSHD handlers each support only one unit. This is documented in Table 1-2 on pages 1-5 and 1-6, but in other places, the documentation implies that more units are supported. If support of more units is required, perform the SYSGEN dialogue to choose the type and number of units to be supported. selecting SYSGEN options matching those of the monitor with which you want to use the handler, and then execute the DEVBLD.COM file to create the new handler.

On page 1-4, in Table 1-1, change the lower memory limit for the BL and SJ monitors from 16 to 12K words. On page 1-7, in section "1.1.2.2 Single-Job Monitor", in the second sentence of the second paragraph, change "16K" to "12K".

On page 2-16, the NOTE is incorrect; replace it with: "A volume containing PIP or DUP can only be removed if the /WAIT or PIP /E or DUP /W options have been used; this is an significant change from earlier versions of RT-11. If you have run PIP or DUP explicitly from a non-system volume and used the PIP /E or DUP /W options, the message "Mount system volume in ddn:; Continue?" generated at the end of an operation really means "Mount PIP/DUP-resident volume; Continue?". If DUP is run from a non-system volume and the system volume is removed. the USR must be SET NOSWAP.

On page 2-20, replace the first paragraph of section "2.5 Mandatory Patches" with the following:

> The RT-11 SR (Software Dispatch Review) and Dispatch kit is shipped as part of your RT-11 software kit. The RT-11 Software Dispatch Review contains operating system and layered product patches that existed at the time of the release of RT-11 V4. The RT-11 Software Dispatch is published monthly; it complements the RT-11 Software Dispatch Review and contains new and revised Software Product Descriptions, programming notes, descriptions of software problems and solutions, documentation corrections, and other information of general interest to RT-11 users. If you have purchased a fully supported (category A) RT-11 system or license you will receive monthly issues of the Dispatch for 90 days after installation of the software. This service can be continued by ordering the RT-11 Binary Program Update Service, or, if available in your area, a DECsupport service, which also provides automatic binary updates of the RT-11 software at no additional cost during the service year.
>
> To determine the mandatory patches you need to install, examine the cumulative index at the rear of the latest issue of the RT-11 Software Dispatch you have received; mandatory patches are flagged with the letter "M". After the installation of your system, you should install the mandatory patches listed in any subsequent issues of the RT-11 Software Dispatch you may receive, in order to ensure that your system software is patched up to current levels.

On page 2-46, in section "2.8.33 Default Device for the EDIT Command", add the following sentence to the end of the first paragraph: "This patch affects only the device on which the monitor looks for EDIT.SAV or TECO.SAV; it does not change the default device for KED or K52."

On page 3-10, the procedure for copying files one at a time from non-bootable volumes will work. however, it IS possible to use the /QUERY option to copy multiple files from a

non-bootable volume. The command format to do this is "COPY/SYSTEM/QUERY/WAIT xx1: xx0:<RET>. The SET USR NOSWAP command is not needed in either case.

On page 8-18, the paragraph which begins "You must respond" is misleading and should be deleted. The second line of the next paragraph is wrong and should be replaced with the the last paragraph of the the RT-11 Installation and System Generation Guide documentation article published in the July 1980 RT-11 Software Dispatch, Sequence number 11.3.1.

RT-11 SYSTEM USER'S GUIDE

On page 1-2, in Table 1, the "Line printer" section is missing the controller (DL(V)11 or PDT-11 printer port) and list of serial hard-copy printer/terminals.

On page 1-3, in section "1.2.1.1 Single-Job (SJ) Monitor, in the first sentence of the second paragraph, change "8K" to "12K".

On page 4-41, before the COPY command /WAIT option description, add the following: "/VERIFY Use this option with the /DEVICE option to verify an image-mode device copy operation."

On page 4-82, in the first line of the EXECUTE command syntax illustration, delete "[/ALLOCATE:size]" following "/EXECUTE:filespec".

On page 4-83, in the first sentence of the /ALLOCATE description, delete "/EXECUTE,".

On page 4-85, in the /LINKLIBRARY option description, delete the three occurences of the word "library".

On pages 4-110 and 8-15, make the following changes to the tables of the default directory sizes used by DUP: Change the line which reads "DY 1" to "DY single density diskette 1", and insert the following line after it: "DY double density diskette 4". Change the line which reads "DL 16" to "DL/RL01 16", and insert the following line after it: "DL/RL02 31". (The method that DUP uses to determine the default number of directory segments is described on page 2-50 of the Installation and System Generation Guide.)

On page 8-4, in the first paragraph of section "8.2.2 Image Copy Option (/I)" delete the phrase "magtape or" from the second sentence which reads: "(This operation is not applicable for magtape or cassette.)".

RT-11 V4
DOCUMENTATION CORRECTIONS AND ADDITIONS

On page 17-8, change the last line on the page to read: "If the terminal is connected via a DZ interface, RESORC displays the current baud rate setting for the terminal, otherwise N/A (not applicable) is displayed.

RT-11 SYSTEM MESSAGE MANUAL

On page 15, add the following NOTE: "The program counter (PC) value after the execution of a HALT instruction is the address of the HALT instruction plus two, thus the PC value after the halts listed in the following sections will be two plus the absolute locations listed."

On pages 15, 16, and 17, add the following new section to the sections on base-line, SJ, and FB/XM monitor halts:

> Absolute location 246 (octal)
>
> The monitor executes this halt if a floating point interrupt occurs under a monitor without floating point support. The program that was executing utilizes floating point instructions and must be run under a monitor with floating point support. (The distributed SJ and FB monitors have floating point support; the BL monitor does not. Floating point support is an optional system generation feature.)

On page 42, replace the corrective procedure paragraph of the "?FORTRAN-F-Dynamic memory overflow" error message description with the following: "Refer to Section 3.2.1 of this manual to increase the memory available to the compiler. If this fails, try compiling the program with threaded code or break up the program into smaller subprograms."

On page 42, add the following to the beginning of the corrective procedure paragraph of the "?FORTRAN-F-Error writing listing file" and "?FORTRAN-F-Error writing object file" error message descriptions: "Usually occurs when there is insufficient free space for the specified file on the output volume; refer to Section 3.1 of this manual. If this fails,"

On page 120, in the corrective procedure paragraph of the "?PIP-W-Reboot" description, add the phrase "or deleted" following the word "moved", and add the following sentence to the end of the paragraph: "Note that the RENAME command can cause deletion of active .SYS files."

RT-11 V4
DOCUMENTATION CORRECTIONS AND ADDITIONS

RT-11 POCKET GUIDE

On pages 10 and 12, in the SKIP option action descriptions, delete the phrase "with SET LP: NOFORM".

On pages 11 and 12, in the SET command LS handler options, underline CR and LC and remove the underlining from NOCR and NOLC.

On page 18, replace the "/H" option description with the following: "When used with /K, reads and rewrites bad blocks; when used with /I, verifies an image mode copy."

On page 84, in the first line of the EXECUTE command syntax illustration, delete "[/ALLOCATE:size]" following "/EXECUTE:filespec".

On page 94, in the LINK command syntax illustration, add "[/ALLOCATE:size]" following "/[NO]EXECUTE[:filespec]".

RT-11 PROGRAMMER'S REFERENCE MANUAL

On page 2-52, in section "2.34 .GTLIN", replace the first sentence of the second paragraph with: "When bit 3 of the Job Status Word is set, if the program issues a .GTLIN request and a line of the form "^C" is the next line in the indirect command file, the program is not aborted; instead, the "^C" causes all subsequent .GTLIN input for that program to be collected from the console terminal, rather than from the indirect command file. When this bit has been set and a line of the form "^C" has been read, if the bit is then cleared and another .GTLIN request is issued, the program will be aborted."

On pages 2-16, 2-21, and 3-17, in the .CSIGEN, .CSISPC, and ICSI sections, add the following sentence to the end of the "cstrng" descriptions: "The command string can not contain any imbedded blanks."

RT-11 SOFTWARE SUPPORT MANUAL

On page 2-7, in the JSW "bit 3" description, replace the second sentence with the same change given above for section "2.34 .GTLIN" of the RT-11 Programmer's Reference Manual.

RT-11 V4
CONFUSING DISPATCH ARTICLE INSTRUCTIONS

All Software Dispatch articles containing patches for RT-11 V4 use the same template text in the instructions for installing the patches. Binary patches to system utility programs and the distributed monitors contain a paragraph which reads as follows:

"This patch is installed using SIPP, the Save Image Patching Program. First, ensure that a copy of the monitor file/xxx.SAV is on a mounted volume. Create the file, xxx.nnn as follows. Replace 'DK:' in the patch below with the name of the device that contains the program file.

```
RUN SIPP
DK:xxx/A/C
   .
   ."
```

The sentence in the paragraph which reads "Create the file, xxx.nnn as follows." could be incorrectly interpreted to mean "Type in the following commands, which will create the file xxx.nnn as the result." The correct interpretation could be stated as follows, which is similar to the instructions given for installing source patches: "Using an editor, create an indirect command file called xxx.nnn. Enter the text below into the file."

MU BASIC-11/RT-11 V2.0
NO RESPONSE TO HELLO COMMAND

The following describes two hardware conditions which will cause MU BASIC to hang at startup time, i.e., conditions which will cause MU BASIC to not respond to the HELLO command.

MINC systems and most 11/03 and 11/23 based Corporate Packaged Systems contain the DLV11-J 4 line serial interface. Loopback connectors which cause transmitted characters to be received as input to the interface for diagnostic purposes are normally left installed on any unconnected lines. MU BASIC sends the message "MU BASIC-11/RT-11 V2 IS OPERATIONAL, PLEASE TYPE IN "HELLO"" to all terminal lines specified in response to the MUCNFG "Enter terminal numbers" prompt. If a line which has a loopback connector installed is specified as one of the terminals, MU BASIC receives back its own startup message as if it were operator input. Since MU BASIC is expecting an HELLO command, it responds with "INVALID ENTRY - PLEASE TYPE IN HELLO", which it receives back as (invalid) input, creating an infinite loop and locking out "real" terminals. Obviously, the cure for this situation is either to remove the loopback connector or to not specify the unconnected line as an MU BASIC terminal.

Another source of a hang at startup time is the lack of line frequency clock interrupts. This can be diagnosed by issuing the RT-11 TIME command; if the time prints as all zeros, the clock is not interrupting. On 11/03 and 11/23 systems, this usually occurs because the power supply-generated LTC signal is not asserting the bus BEVENT line. (On the older 11V03 systems, the front panel LTC switch must be in the ON position.) To enable the clock on a PDT-11/150 system, remove the PDT top cover by removing the two screws at the upper left and upper right of the back of the PDT cabinet. A small rectangular pack of switches will be found on the top of one of the PDT modules; switch number 2 should be moved from the ON to the OFF position.

If MU BASIC suddenly fails to respond to terminal input after it has been running for awhile, it may be that the line frequency clock interrupt enable bit has become cleared. To correct this, halt the CPU, deposit 100 in location 777546 by using uODT or the console switches/keypad. and restart the CPU. (For 11/03 and 11/23 systems without the BDV11 bootstrap module, the above procedure is irrelevant since these systems do not normally have a line frequency clock status reqister; a "?" will be printed by uODT when attempting to open location 777546.)

Finally, note that the HELLO command cannot be typed-ahead; the message "MU BASIC-11/RT-11 V2 IS OPERATIONAL., PLEASE TYPE IN "HELLO"." must appear before the HELLO command will be recognized.

PDP-11 FORTRAN IV
ASIN AND ACOS FUNCTIONS

The FORTRAN IV OTS library (RT-11, RSTS/E and/or IAS/RSX) does not contain the ASIN and ACOS functions; the FORTRAN IV-PLUS OTS library does. The following functions were written based on the algorithms used in FORTRAN IV-PLUS, as shown on page B-1 of the PDP-11 FORTRAN IV-PLUS User's Guide.

```
      FUNCTION ASIN(X)
      IF(X.NE.-1.)GO TO 10
      ASIN=-3.14159265/2
      RETURN
10    IF(X.NE.0)GO TO 20
      ASIN=0.
      RETURN
20    IF(X.NE.1.)GO TO 30
      ASIN=3.14159265/2
      RETURN
30    ASIN=ATAN(X/SQRT(1.-X**2))
      RETURN
      END


      FUNCTION ACOS(X)
      IF(X)10,20,30
10    IF(X.GT.-1.)GO TO 15
      ACOS=3.14159265
      RETURN
15    ACOS=ATAN(SQRT(1.-X**2)/X)+3.14159265
      RETURN
20    ACOS=3.14159265/2
      RETURN
30    IF(X.EQ.1.)GO TO 35
      ACOS=ATAN(SQRT(1.-X**2)/X)
      RETURN
35    ACOS=0.
      RETURN
      END
```

11/03, 11/23 SYSTEMS
LSI-11 BUS MODULE FACTORY CONFIGURATION

Several LSI-11 bus modules are manufactured with certain default factory jumper/switch settings, some of which are not optimal for certain applications. These modules include the BDV11 bootstrap/terminator module, the DLV11 family of asynchronous serial interfaces, the MSV11-DD 32KW MOS memory module, and the MXV11-A memory and dual asynchronous serial interface module. The BDV11, the DLV11-J, and the MSV11-DD modules are incorporated into all 11/23 Corporate Packaged System configurations, and in the newer 11/03-L based Corporate Packaged Systems (not in the older 11V03 or DECLAB-03 configurations) without having some of these jumper/switch settings changed; although some of these settings are changed in MINC-11 and DECLAB-11/MNC systems, I am not sure whether they are changed in DECdatasystem configurations. This article describes some of those jumper/switch settings and their effects; for information on how to change the jumper/switch settings, refer to the new Microcomputer Processor Handbook and Microcomputer Interfaces Handbook, or the older Memories and Peripherals Handbook.

(1) The MSV11-DD 32KW memory module has a jumper option which allows access to 30KW (as opposed to 28KW) of memory on 11/03 systems and in unmapped mode on 11/23 systems; this option is referred to as "bank 7 addressing". The additional 2K words of memory provided by this option can be of particular advantage in certain applications, such as MU BASIC-11/RT-11, FMS-11/RT-11/ARTS, and RSX-11S. This option can be utilized as long as there are no installed interfaces whose I/O page addresses are in the 28-30K range of I/O page addresses on 11/03 systems (addresses 160000-170000), and in the corresponding 124-126KW range of I/O page addresses on 11/23 systems (addresses 760000-770000). The IBV11-A, DRV11, DUV11, DZV11, and additional RLV11 controllers are normally installed with I/O page addresses in the above range. If there are interfaces whose I/O page addresses are in the above range and it is desired to use the bank 7 addressing option, the interfaces can normally be reconfigured so that their I/O page addresses are above the specified range.

(2) The BDV11 bootstrap/terminator module is shipped so that it prints the "START?" prompt to allow operator specification of the bootstrap device and unit. Switch settings on the BDV11 allow for auto-booting of unit 0 of a specific device on power-up or when the console panel RESTART switch is activated. This is often desirable in certain applications to minimize operator interaction and to allow for power fail recovery.

(3) The DLV11, DLV11-F and DLV11-J family of asynchronous serial interfaces and the DLV-compatible serial interfaces on the MXV11-A module are manufactured with an option called "halt on framing error" enabled. (The DLV11-E module also has this option, but it is manufactured with the option disabled.) A framing error is generated when a received character does not have a stop bit, such as when the BREAK key is struck. This option was designed to allow the BREAK key on the console terminal to halt the CPU thereby allowing the user to gain entry to console ODT, normally for debugging purposes. Although this option is valuable on development systems, having it enabled on production systems is usually not desirable since accidental hitting of the BREAK key would cause a system halt. It is also important to disable this option on any non-console DLV interfaces installed, since a powered-down piece of equipment connected to one of these interfaces can also generate a framing error condition which might otherwise halt the CPU. Note that if the system has a console control panel, disabling the halt on framing error option does not remove the ability of the operator to halt the CPU and enter console ODT, since this ability is available through the console ENABLE/HALT switch.

Dear Mr. Demers:

In an effort to keep the DECUS users in your SIG up-to-date on the latest training available from Digital, Educational Services has agreed to send

What When Where

is available on training to each SIG Newsletter Editor.

Enclosed with this letter is a standard copy of "The DIGEST: Training Update & Schedules", published quarterly by Educational Services. It contains all of the current training schedules and news pertinent to the U. S.

This information is purely for the benefit of the SIG members and will contain neither prices nor advertising. If the reader wants prices and descriptive literature, he can write or contact his nearest training center.

As well, from time to time, we wll submit abtracts of seminars held in various cities that may be of interest to your members. If you agree, feel free to publish this information.

If you have any suggestions on how you would like this information displayed, please let us know. Generally the schedule pertaining to ONLY your SIG would be submitted to your Newsletter in "camera ready" (photostat) copy on 8 1/2 x 11 paper with one inch margins. It will arrive on your desk approximately the first of each quarter throughout the year. We would think that your SIG members would want to see this schedule in your next Newsletter.

Yours sincerely,

Judy Arsenault

Judy Arsenault
SUG Administrator
DECUS

J. E. Richardson
Manager
Technical Seminars
Digital Equipment
Corporation

RT-11 CUSTOMER TRAINING PROGRAM

SAN FRANCISCO (408) 984-0200 Ext. 2142

Course #	Title	Length	Start Day	Tuition	Training Credit	July				August				Sept.					October				November				December				
						7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29
EY-J2114-AO	Introduction to Minicomputers A/V	5		$ 400	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2016-AO	Introduction to Minicomputers	5	MO	520	1	●					●					●						●					●				
EY-J2116-AO	Introduction to the PDP-11 A/V	5		450	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2024-AO	PDP-11 Assembly Language Programming	5	MO	580	1		●			●		●			●		●			●			●		●			●			
EY-J2050-AO	Programming in MACRO-11	5	MO	630	1			●					●			●					●			●					●		
EY-J2048-AO	PDP-11 Concepts (for High-Level Prog.)	3		460	1																										
EY-J2000-AO	RT-11 User	5	MO	630	1	●						●							●					●							
EY-J0016-AO	Programming in FORTRAN IV	5	MO	630	1					●													●								
EY-J2002-AO	RT-11 Programmer	5	MO	630	1		●						●							●					●						
EY-J2004-AO	RT-11 System Programmer	3	MO	520	1										●													●			

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

						7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29
EY-J2114-AO	Introduction to Minicomputers A/V	5		$ 400	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2016-AO	Introduction to Minicomputers	5	MO	520	1				●						●			●				●						●			
EY-J2116-AO	Introduction to the PDP-11 A/V	5		450	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2024-AO	PDP-11 Assembly Language Programming	5	MO	580	1	●		●		●		●			●		●		●		●		●		●		●		●		
EY-J2050-AO	Programming in MACRO-11	5	MO	630	1		●		●				●			●				●		●		●				●			
EY-J2048-AO	PDP-11 Concepts (for High-Level Prog.)	3	MO	460	1						●							●							●						
EY-J2000-AO	RT-11 User	5	MO	630	1					●									●												
EY-J0016-AO	Programming in FORTRAN IV	5	MO	630	1		●						●					●									●				
EY-J2002-AO	RT-11 Programmer	5	MO	630	1	●														●											
EY-J2004-AO	RT-11 System Programmer	3		520	1																										

NEW YORK (212) 971-3545

						7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29
EY-J2114-AO	Introduction to Minicomputers A/V	5		$ 400	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2016-AO	Introduction to Minicomputers	5	MO	520	1			●					●				●				●						●				
EY-J2116-AO	Introduction to the PDP-11 A/V	5		450	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2024-AO	PDP-11 Assembly Language Programming	5	MO	580	1	●			●			●						●			●						●				
EY-J2050-AO	Programming in MACRO-11	5	MO	630	1		●			●							●		●			●						●			
EY-J2048-AO	PDP-11 Concepts (for High-Level Prog.)	3		460	1																										
EY-J2000-AO	RT-11 User	5	MO	630	1						●								●												
EY-J0016-AO	Programming in FORTRAN IV	5	MO	630	1		●					●						●											●		
EY-J2002-AO	RT-11 Programmer	5	MO	630	1																										
EY-J2004-AO	RT-11 System Programmer	3		520	1																										

RT-11 CUSTOMER 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	July				August				Sept.					October				November				December				
						7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29
EY-J2114-AO	Introduction to Minicomputers A/V	5		$ 400	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2016-AO	Introduction to Minicomputers	5	MO	520	1			●					●				●					●			●				●		
EY-J2116-AO	Introduction to the PDP-11 A/V	5		450	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2024-AO	PDP-11 Assembly Language Programming	5	MO	580	1	●			●			●					●		●		●			●			●				
EY-J2050-AO	Programming in MACRO-11	5	MO	630	1		●			●			●					●		●		●			●			●			
EY-J2048-AO	PDP-11 Concepts (for High-Level Prog.)	3		460	1																										
EY-J2000-AO	RT-11 User	5	MO	630	1				●						●																
EY-J0016-AO	Programming in FORTRAN IV	5	MO	630	1			●								●					●							●			
EY-J2002-AO	RT-11 Programmer	5	MO	630	1					●						●															
EY-J2004-AO	RT-11 System Programmer	3		520	1																										

BOSTON (617) 275-5000 Ext. 2380

						7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29
EY-J2114-AO	Introduction to Minicomputers A/V	5		$ 400	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2016-AO	Introduction to Minicomputers	5	MO	520	1		●		●		●		●			●			●			●			●		●		●		
EY-J2116-AO	Introduction to the PDP-11 A/V	5		450	1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
EY-J2024-AO	PDP-11 Assembly Language Programming	5	MO	580	1	●		●		●		●			●		●		●		●		●		●				●		
EY-J2050-AO	Programming in MACRO-11	5	MO	630	1	●		●		●	●		●		●	●	●		●	●			●	●	●		●	●			
EY-J2048-AO	PDP-11 Concepts (for High-Level Prog.)	3	MO	460	1	●										●								●					●		
EY-J2000-AO	RT-11 User	5	MO	630	1		●						●			●				●			●					●			
EY-J0016-AO	Programming in FORTRAN IV	5	MO	630	1			●									●				●						●				
EY-J2002-AO	RT-11 Programmer	5	MO	630	1			●									●				●								●		
EY-J2004-AO	RT-11 System Programmer	3	WE	520	1				●													●									

[illegible]

I would like to have some information regarding Symposia tape copies. The reason that I did not send this letter to one of the people mentioned in the June Minitasker is because they didn't provide compatibility with the media which we use. Our system uses cassettes and RK05's as magnetic media; I was wondering (hoping) if you would be able to copy onto cassettes instead of requiring magtape.

The files which I would like to get are:
The RATFOR translator sources (& documentation, etc.), from the Spring-'80 symposium; and, form the Fall-'79 symposium: TECO files, OTHELLO sources, BASIC Cross-Reference programs, and FORTRAN Cross-Reference program.

Also, I would like to know if it would be possible to get an overlaid version of ADVENTURE: We use a Version 2 system, with 16K-words of core, and the available version supposedly requires 28K-words of core.

Please tell me whether or not it is possible to have those files copied onto cassette; and if so, how.

Thank-you

Keith Gregory
29 Franklin Ave.
Swampscott, MA. 01907

ALBERT EINSTEIN COLLEGE OF MEDICINE
YESHIVA UNIVERSITY
EASTCHESTER ROAD AND MORRIS PARK AVENUE
BRONX, N.Y. 10461

Dear Mr. Demers:

We are presently working under RT-11 with a VT55 terminal with graphs plotted on a VR17 screen, off a VT11 graphics processor. We need a hard copy of the image on the VR17 screen. We hope one of your readers may offer a suggestion on what software and/or hardware is available to achieve this. In particular, we would appreciate information on any programming effort to scan the display file created with DEC graphic-11 and recreate the display on the VT55 screen.

Sincerely,

A. Ahuja

Arun A. Ahuja
Bioengineer

DEPARTMENT OF SURGERY
Tel. (212) 430-2262

[illegible]

Ned W. Rhodes
DTNSRDC
Code 2950
Bethesda, Md. 20084
(202) 227-1592
23 July 1980

Mark Meinrath
Instructor
Wake Forest University
Winston-Salem, NC 27109

Mark:

I just saw your letter in the most recent issue of the MiniTasker (June 1980--Vol. 6 No. 3). Your TECO problem is an easy one to solve. Although you didn't mention which operating system you are using, I am sure that it is F/B. Insert the statement:

```
SET TT NOCRLF
```

into your start-up indirect command file, or type it in by hand.

The random insertions of line feeds and the like will never bother you again. I am sure that you have noticed that random insertions are not a problem with the S/J monitor, only with the F/B (maybe the X/M) monitor.

Also, I find that it is easier to start TECO with:

```
SET EDIT TECO
EDIT/EXEC VEG
```

rather than the method you used in the example you submitted to the MiniTasker.

The same solution was also submitted by:

Gary Beckmann
Gary Beckmann
Joint Center for Radiation Therapy
50 Binney Street
Boston, MA 02115

Standard Telecommunication Laboratories Limited

London Road, Harlow, Essex, England CM17 9NA
Telephone Harlow 29531
STD Code 0279
Telex 81151

Dear Mr. Demers, Re: Your letter to The Mini-Tasker, April 1980

As far as I know we were the first people to draw the problem to DEC's attention. The solution was found by our local DEC office software support man and should have been passed into the DEC system. If you need any further information, please contact me. We have several ALR's available from MUBASIC V2 which use the technique established and have used it for about one year.

This Refers to the user request by J.P. Farine on page 14 of the April MiniTasker.

Yours sincerely,

John Lee
J. E. Lee
R672 Dept.

Roland R. Burgess,
Senior Analyst,
ICI Americas Inc,
Wilmington, Deleware 19897

This is a reply to the User Request in MiniTasker V6N3, June 1980, regarding FMS-11 and use of double height/double width letters. To clarify the situation, DEC strongly refuses to market the FMS sources, so we have not modified FMS-11 itself. Instead we chose to modify forms USING the Forms Editor that is part of FMS. Realize that this is device dependant, but if you are willing to live with that, you can use our method. (To follow this you should know the specifics of VT-100 control character sequences.) All you need to do is get the escape sequences into a form once; from there you can use FED to edit new forms from this patched form. You make a form called "GRAFIX" or some such name, with only this text;

```
?#3Double Height
?#4Double Height
?#6Double Width
?#5Normal Again
```

When you have saved this form, DUMP it to find out where the question marks appear, and PATCH it to change them to escapes. The escape should be <233> rather than <033> so it will work on things like the RSTS/E emulator.

Of course the same approach could be used for other attributes like graphics. I expect to be showing some examples of our screens at the November DECUS. Our approach works now; whether it will continue to work with future releases of FMS-11 is moot. The FMS-11 group has said informally that they are looking into supporting graphics and double width (but not double height, presumably because of VT52 compatability). Despite these minor limitations, we have found FMS-11 to be a very serviceable product.

Hope this information is of help.

Sincerely,

Ian F. D
Ian F. Darwin
Univ of Toronto Computing Services
10 King's College Road
Toronto, Ontario M5S 1A1

A Comparison Between REMOTE-11 and STAR-eleven

Howard Schultens
Physiologisches Institut 2
Humboldtallee 7
3400 Goettingen, W. Germany

REMOTE-11 was designed as a multi-user editor with a capability of down-line-loading to computers connected over serial lines. REMOTE was written in 1974/75 by Rick Hully -- apparently almost singlehandedly, looking at the sources -- to provide a layered enhancement to RT-11 for users who were beginning to try to connect one or two additional terminals to an RT-11 system. Since some people wanted to connect up dedicated processors to an RT-11 system, REMOTE provides on-line support of a subset of the RT-11 programmed requests so that basic console and file operations (.LOOKUP, .ENTER, .READW, .WRITW, .CLOSE, .DELETE, .CSISPC) together with the most necesssary job control requests (.INTEN, .SYNCH, .EXIT, .SETTOP, .HRESET, .TRPSET, .GTJB) can be executed by a program in a very minimal PDP-11 running as a satellite. (This is not a complete list of requests supported by REMOTE). REMOTE implements these requests and the down-line-loading by the use of Digital Data Communications Message Protocol (DDCMP) messages to and from the host computer. This protocol provides not only reliable support for message transmission over error-prone lines, but it allows, in principle, a portability within DECnet systems, since the protocol is supposed to be independent of the processors involved. The overhead for this protocol is suprisingly high if it is not hardware-supported with a KG11-A: it is equivalent to sending the same message around 30 times. Since there is no equivalent to the KG11 for LSI's, this can be an important timing factor. PDP-11's can be connected to one another via telephone and microwave channels using this protocol, but one asks if DECnet compatibility is really necessary at this level -- especially in view of the lack of uniform standards just within RT-11 itself.

We purchased REMOTE-11 in 1976 to provide RT-11 support for a number of stripped-down PDP-11/03 satellite computers doing data aquisition in the laboratory. At the time it was the only package available for a low-cost "local" computer network using standard interfaces. The programmed requests supported by REMOTE enable satellites to run FORTRAN programs which have been linked with the support module RTSIM; this made it an ideal prospect for software support in the laboratory where one often needs the speed and reliability of a dedicated processor, but complete computer systems for each lab are financially out of the question. As it was delivered, REMOTE did not work. After we ourselves installed the patches from the Software Dispatch, it ran fairly well, but we were continually rebooting the host because of errors we could not localize. DEC provided us with no support except the patches that appeared in Software Dispatch, Digital Software News, etc. (what is the difference between these publications?). And when we went over to RT-11 V03, REMOTE no longer supported FORTRAN programs. These may be problems which everyone has with a new software system; in fact they have nothing to do with the important, basic features of our situation, namely the most effective, local coupling of a PDP-11 computer to another PDP-11 computer. We moved away from REMOTE in 1979 for three reasons:

First, the large amount of program development we were doing necessitated the support of .SAV and overlayed files on the satellites. REMOTE can pass the host background to any terminal in the network to allow the use of .SAV files (that means any systems programs), but with 3 or 4 users, this becomes a serious bottleneck. The host background becomes smaller with each additional user (10K left on a 28K machine with 4 satellites), so that many .SAV files no longer can run. STAR-eleven provides us with a complete RT-11 SJ environment on the satellites, so that all program development can be done on the satellites. In addition, commercial software for which the .OBJ files are not supplied will only run on the host under REMOTE; it runs unchanged on the satellites under STAR-eleven.

Second, adequate software support and specialized hardware consultation are necessary, since local computer network technology is a new field. As a university institute, we must keep the time devoted to systems programming and hardware development to a minimum. STAR-eleven provided us with interfaces and cabling specifications that eliminated most problems; the remaining problems were how to get an IBM Selectric, a refrigerator, or fluorescent lighting onto another phase.

The third reason is speed: the host connections run at 300 KBaud in STAR-eleven and the software can keep up with it. As mentioned above, the DDCMP protocol in REMOTE using 9600 Baud serial lines is extremely slow. STAR-eleven uses a much faster protocol on these faster interfaces. We were first hesitant at the extra expense of the WB(V)-11 interfaces, but the investment is thoroughly worth it. This opens a whole new area of interactive data acquisition and processing in the laboratory.

REMOTE-11 is primarily terminal-oriented, with the down-line-loading and runtime RT-11 support added almost as an afterthought (isn't RTSIM an adaptation of the FORTRAN standalone SIMRT module?) Probably the software groups at DEC were taken by surprise nearly as much as the users were by the demand for and versatility of the inexpensive PDP-11/03. Attaching a processor with the same command structure to an RT-11 host automatically leads to demands for as many host services as possible at the satellite. By relinquishing the demand for DECnet protocols within a local network, a great increase in software power can be achieved at very modest cost.

Appended here is a table giving a brief overview of our institute computer network. We moved the VT-11 Display to one of the satellites to take advantage of the much larger user space available there.

Host: PDP-11/34
64KB main memory
2 x RK05
64 MByte Disk (supported by STAR-eleven)
Magtape (supported by STAR-eleven)
300 LPM Printer (STAR-eleven has a spooling facility)
Plotter (supported in a natural way by STAR-eleven)
WB-11 Satellite interfaces

Satellites: 1: PDP-11/03 (soon to be 11/23): Data Processing Node.
64KB main memory
UNIBUS Converter
VT11-A Display Processor (**)
LA36 DECwriter (**)
WBV11 Host interface

Nodes 2 to 8 are all similar:
64KB main memory
VT52 (or equivalent)
Analog-to-digital converter
100KHz Clock
DRV11 16 bit parallel i/o
WBV11 Host interface

One satellite has floppy disks and can run as a stand-alone system.

The comparison with a single large minicomputer supporting equivalent hardware is instructive: We have 8 users operating in 224KWords of main memory, each with fully isolated user partitions and low operating system overhead.

(**) STAR-eleven supports the VT-11 and LA36 as combined console display and keyboard just as RT-11 does, with CTRL/E to turn printing on and off.

Review of a Laboratory Computer Installation

Dieter Michael
Max Planck Institute for Biophysical Chemistry
D-3400 Goettingen, West Germany

In the late 60's I started working with computers in the fieid of on-line data-acquisition for neurobiological applications. In those days computers were hardware-only packages, and not even a body of experience was available to the application programmer. At that time we saw our task as offering the power of the hardware to the user, as a practical tool, to satisfy the users needs. We recognised that we could satisfy users needs most efficiently with two classes of experimental service:

(1) Open loop experiment - Allowing the experimentor to work on-line with the computer by means in watching the performance of the data acquisition procedures on a graphic display, to be able to modify experimental parameters interactively via a keyboard or potentiometers etc.

(2) Closed loop experiment - In a more complex case, the computer itself performs the task of modifying the experimental parameters automatically.

In 1968 we installed a DEC LINC-8 system, with 4kw memory and DECtapes. At this time the LINC-8 was the only system which handled all the necessary A-D conversion, display and external sense lines etc. for running a real-time experiment. The lab was located 20 meters from the computer room, and in the first stage an operator, in the computer room, monitored the scope, and communicated with the lab via an intercom. This was inadequate, and we installed a parallel scope and console in the lab so that the experimentors themselves could do the interaction.

With time the number of users and labs grew, a single computer couldn't handle the experimental load. We enlarged our computing facilities by installing a PDP-12, DEC's replacement for the LINC-8. Now two on-line experiments could run simultaneously, but software development, off-line data analysis (statistics, plotting) usually claimed one machine. So we installed a second PDP-12, and dedicated the LINC-8 to offline work.

These computer-systems were limited by both hardware and software technology, and aimed simply to satisfy users needs as directly as possible. All labs had to share the two PDP-12's - and these huge machines were not portable. Thus, all signals had to be fed between lab and computer room. We built two portable 'terminals' consisting of teletype, VR-12 point plot scope and connections for the relays and analog channels. These 'terminals' were connected to the computer with a cumbersome and expensive, 62 wire cable, over a distance of upto 60 meters.

In the mean time our department had grown even more, to twelve experimental groups. It was the third time we had outgrown the computers capabilities. Computer time became so valuable that meetings had to be held to establish user priorities.

- A new system

It was time to look for a system which would handle both our current and projected future needs. We decided to use the lessons we had learnt with the PDP-12's, and set up a new system. We defined the following goals:

(1) To support upto four independent computer working places.

(2) To supply users with computing facilities as close as possible to the experimental setup (the computer should come to the user).

(3) Not to be limited by short distances or expensive cables.

(4) To provide a system which would be both: flexible and growth oriented (we could not afford spending money for replacing our computer system every three years).

(5) To have a system which would be easy to use (even for medical scientists) and reliable (to keep the users happy).

We decided quite early to use PDP-11's, VT-11 graphic processors, and Laboratory Peripheral Systems (LPS).

We evaluated various PDP-11 operating system alternatives. A multi-user system, like RSX-11, did not solve the main problems, and presented further problems of connecting several VT11's and LPS's to a single UNIBUS. RT-11's simplicity and flexibility attracted us - and it was the only other alternative at the time.

It was clear that each lab required a computer with console, display and LPS. Supplying each laboratory with a single-user computer, plus mass storage, would have been easy, but expensive. This solution would also deny the users a comfortable method of handling off-line work, like printing and plotting, and software maintenance. The only way to make these facilities available to the lab computers, was by centralising the peripherals and supplying the remote lab computers with access to peripherals via communication lines and the appropriate software.

We installed a PDP-11/40 and a GT-40 (PDP-11/10 with VT-11), and linked them together with a pair of DL11/E's. Our first approach was to enable down-line loading of application programs to the 11/10 satellite, and to store the experimental data from the satellite. Around this time REMOTE/RT-11 was announced. We liked the idea of REMOTE, but it only seemed to go half way towards providing a complete RT-11 service to satellite users. REMOTE did not supply an RT-11 user interface and only supported a small subset of the RT-11 program requests. We thought it should be possible to get a full satellite emulation of RT-11 up and running.

The resulting host/satellite system, 'STAR-eleven', is described in detail in [1]. The first implementation was running after two months. The evolution and growth of STAR-eleven has continued to satisfy the expanding needs of the department.

Our present configuration, shown in figure 1, serves the needs of 16 labs with 30 users. The central host computer is located on the ground floor of the department, the labs are mostly on the other two levels. The total system can be described as a parallel processing system with nine processors, 288k words of memory, supporting five VT-11's, four VT-100's, four LPS's, four EAE's, four FIS's, four RK05's, one RL01, four floppies, magtape, papertape, three plotters and a high quality typewriter.

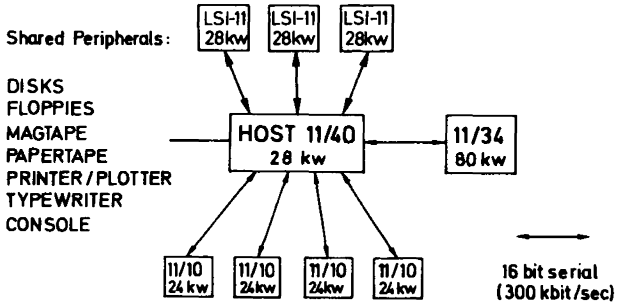


Figure 1. The present configuration.

There are four portable satellites (PDP-11/10, 24kw, VT-11, LPS, LK-40 keyboard). Each lab has a plug, which can be connected to any of the four portable satellites. These plug connections lead to a distribution panel in the computer room, which allows each satellite to be connected to any lab.

Three non-portable satellites (LSI-11/2, 30k, VT-100) in the user room are used mainly for: program development, data analysis, plotting, printing, text editing and formatting, and for debugging new software and hardware.

The fourth non-portable satellite (11/34, 80k, VT-11) in the computer room provides a very high quality working place for large FORTRAN jobs, and other jobs where memory space and i/o speed are critical. If this satellite is not in use it serves a remote command file handler for other satellites.

The present system has more than satisfied the goals we set: The system has grown organically (growth was oriented by the users needs) and now it supports twice the number of satellites initially planned for. Users, programmers and hardware development groups no longer need to compete for machine time. Satellites may be located in the laboratory, close to the experiment, and thus our distance and cable goals have been satisfied.

The system is very easy to maintain, and no longer requires specialist supervision. An operator has the daily and semi-automated task of rolling out the previous days data. Apart from the this task the system can run unsupervised for weeks at a time. The identical structure of all laboratory satellites greatly simplifies the task of providing a suite of applications programs.

It is extremely important to users that experiments be not aborted due to computer system failures. If a laboratory satellite breaks down it can be quickly replaced with one of the other satellites. With a little more work, a satellite can also replace, or provide spare parts, for a failed host (very rare). The replication of hardware provides the redundancy required for running a system reliable.

With the PDP-12's, the computer system was the center of a departmental conflict, and dictated the daily routines of the department. With the present system, users treat the satellite as just another of the practical tools available for laboratory work, and do not even notice the SYSTEM behind it.

In our opinion, only multi-computer systems provide the reliability, flexibility and strength required to support all the tasks involved in keeping the users happy in a large, diverse experimental research department.

- Summary of systems tasks

Having a flexible and expandable system has meant, by parkinsons law, that our system has continued to grow by adding services to satisfy users needs and to provide all personal with sufficient machine time.

The 'portable satellite' concept allows all user classes to use the computer in the appropriate place, that is, in the laboratory for experiments, in the computer user room for program development, or in the hardware shops for the development of new laboratory hardware and maintenance.

It's easy to under-estimate the number of tasks involved in the complete support of a large department. The following list briefly outlines most the major areas of support, and compares the present systems performance with the performance of our previous systems.

Providing sufficient computer time and space:
System runs 24 hours a day, 7 days a week.
Supports 4 real-time experiments and 4 offline tasks simultaneously.
Satellites are replacable if hardware fails.
Satellites provide between 21k and 27k words program space.
[previously fights for computer time]

Providing an easily used computer system:
Satellites are 'turn-key' bootstrapped.
Satellites provide a complete RT-11 working place.
User transparent software/hardware matching.
Total system requires only one hours attention per day (see below).
[previously cumbersome and unreliable, required specialist supervision]

Running a variety of simultaneous real-time experiments:
A suite of application programs for a variety of applications.
(See below for a list of application areas.)

Data management and archiving:
Automatic application suite file-naming and archiving of data.
System operator spends an hour a day archiving previous days data.
[previously required user attention]

Offering the major computer languages:
MACRO, BASIC, FORTRAN, FOCAL, PASCAL.
[previously no operating system, and only assembler and FOCAL]

Maintenance:
Satellite maintenance does not affect network.
Host maintenance is rare, and planned (PM).
[previously down once a month]

Text preparation:
Window editors provide easy-to-use editing environment.
RUNOFF and high quality typewriter for first drafts.
[previously almost nothing]

Program development:
Each satellite can be used for program development.
Programs can be tested on the machines they are destined for.
[previously competed for computer time with experiments]

Data analysis:
Users to write analysis programs in their favorite language.
SSP, LSP and BMDP subroutine libraries.
[previously crude analysis, mostly at local computer center]

Plotting and printing:
Line printer spooler.
HP Four colour plotter, Benson Electrostatic printer/plotter.
Plotter subroutine libraries.
[previously competed with programming and experiments]

Supporting laboratory hardware:
Laboratory hardware can be easily connected to portable satellites.
Satellites remove the requirement for 'laboratory preprocessors' and host 'front-end processors'.
New hardware can be developed and tested on satellites without interrupting network operations.
[previously required large, cumbersome, expensive 'terminal']

Interfacing to software packages:
Using an established and popular operating system (RT-11) provides access to a large number of software packages (BMDP, DECUS etc.).
[previously no problem, i.e., no packages]

Integrating computer purchases and supplies
Users dont have to worry about disks and so on.

- Summary of the major application areas

(1) Neuroendocrinology - Evaluation of Radioimmunoassay data of hormones and of radioenzymatic monoamine determinations and their statistical analysis.

(2) Somatosensory system - Collecting two simultaneous Peri-Stimulus-Time Histograms (PSTH) for i/o correlation of single thalamic neurons after quantitative mechanical stimuli of the face skin in the cat.

(3) Auditory system - Digital signal processing of natural sounds and speech signals (filtering, exponential weighting of the envelope, spectral and temporal modifications). The modified signals are presented as stimuli in neurophysiological and psycho-physical on-line experiments.

(3) Visual system - Quantative analysis of visual function with stimuli of different colours and shapes. The satellite computer collects PSTH's of spike responses, and on-line cross-correlates ongoing spike trains and controls an oscilloscope display used as visual stimulus.

(4) Functional topography of the visual cortex - Closed-loop experiment that sequentially collects PSTH's from 14 different orientations. The computer controls the visual stimuli (speed, angle, increment etc.).

(5) Visual neurophysiology - Closed loop experiment collecting neuronal discharge time intervals, and controlling optical stimulator for moving turning and flashing slide stimuli. This experiment has these applications: (a) Fast determination of orientation specificity of neurones. (b) Scanning complex optical patterns for neuronal transfer properties. (c) Analysis of spatial/temporal interaction.

(6) Visual neurophysiology - Two stage experiment - Firstly uses average evoked potentials to establish nuclei identification and input components in the Columbia Livia of the Pidgeon. Secondly, collects fast, averaged, PSTH's.

(7) Neuroanatomy - A Zeiss microscope with joystick is used to trace the dye-injected cellular boundaries. The computer monitors, and stores, the trace path x/y coordinates. The resulting reconstructions are investigated for cell positions, diameters, surfaces, and dendritic distributions.

(8) Neuroanatomy - The semi-automatic image analysis system (Leitz ASM) is connected to the network as a satellite. The X/Y/Z coordinates of serial brain sections and brain structures are stored and used for three-dimensional reconstruction of the brain.

(9) Membrane biology - Evaluation of time constant and amplitude of the autocovariance function of conductance fluctuations of membranes. The satellite is connected to Honeywell Correlation and Probability Analyzer.

(10) Automatic EEG analysis - The different patterns of EEG's are segmented using digital signal processing techniques. The segmented data are further classified (cluster analysis) and presented in a form suitable for aiding evaluation of clinical EEG's.

- Acknowledgements

I would like to thank Ian Hammond and Werner Buchholz for their teamwork in getting STAR-eleven on the air, and the users of the Department of Neurobiology for their cooperation.

- References

[1] Hammond,I.; The STAR-eleven general-purpose multi-computer system.
in this issue of the MINI-tasker

[2] Hammond,I.; Michael,D.; Buchholz,W.
An approach to stuctured systems.
DECUS Europe proceedings, Monte Carlo, Monaco - 1979

The Satellite Connection

Werner Buchholz, Dieter Michael and Ian Hammond
BUCHHOLZ-hardware, Nienkemperstr. 30
D-1000 Berlin 37, West Germany

Connecting computers together reliably at high speeds, and at low cost, is not a simple task. This paper discusses a series of different solutions to local network connections over a period of four years.

One preliminary point should be made about the differences between local and global intra-computer connections. Global computer connections over long distances and public lines are, a priori, slow and error prone. However, connections over short distances and private lines can utilize techniques that are not available to global connections, to produce fast and reliable intra-computer connections. The paper outlines some of these techniques.

- 8 bit serial connections

The first connection was a straight console-type DL11/E interface. This had a baud-rate of 9600, which was inadequate for the communications task at hand. Our first modifications consisted of modifying the frequency divider to get a rate of 19.2k baud, and then changing the quartz crystals for a rate of 38.4k baud.

We immediately got problems of data overrun. It was possible for RT-11 to block interrupts for periods of over a millisecond, which limits any non DMA devices to a transfer rate of 9600 baud. The DL11/E had some unused lines on it. The solution was to use these lines to implement an 'interlock'. 'Interlock' simply means that a sender waits for receiver to signal receipt of a character before sending the next character. 'Interlocked' transceivers can send and receive at a maximum rate.

Our first interlock implementation was handled in software; whenever a receiver picked up a character it would complement the state of the 'interlock line'. This complement operation would cause a 'dataset' interrupt in the senders machine. The sender would interpret this dataset interrupt as a receipt signal.

The software 'interlock' worked quite reliably but required considerable interpretation in the interrupt handlers, and thus slowed down transmission rates. So we moved the whole interlock operation into hardware. This was done using a variety of tricks, including piggy-back chips and off-board logic. However it worked, and provided an effective thruput rate of 5k words per second. Implementing the interlock in hardware eased the software workload considerably, and the average interrupt service routine was reduced to only five instructions.

Clearing up a large problem often exposes smaller ones. The next problems that showed thru were distance, galvanic isolation and distortion. Supplying line drivers allowed us to send signals over distances of 200 meters. Optical isolated receivers (opto-couplers) solved the galvanic isolation problem. The technique of differentially driven signals over twisted pair cables cleared up the problems of externally induced distortions.

- 16 bit parallel connections

The DL11/E transfers a byte at a time, and we figured that a word transfer device would give us higher speeds (particularly because it would halve the number of interrupt service routines required). So, the next step was to modify a 16 bit parallel DR-11 interface for hardware interlock. This provided us with very high speeds, but with a problem: The cables we used were all twisted pairs, and, for 16 bits data plus 4 control lines, in two directions, we required 40 twisted pairs. This is expensive in terms of opto-couplers and line-drivers, and over distances of 200 meters the cable itself is expensive.

The two resulting interfaces, serial and parallel, provided the reliability and range of speed we required, and they were not too expensive, but they were dirty. When six satellites were connected, the bus extension box looked like a mad scientists dream: Piggy back chips, cut etches, patch wires and extra boards to handle the opto-couplers and differential drivers.

- 16 bit serial connections

The final stage consisted of integrating all the lessons we had learned and producing, from scratch, a clean printed circuit board to carry all the logic required for an intra-computer connection.

Some networks use DMA based intra-computer connections, the reasoning why we didnt, follows: In the end, the disk seek and transfer times are the limiting factors for our application, and the devices described here already provide more power than the software can utilise. Secondly, our relative efficiency is improved by reduced software overheads, i.e., the highly reliable connection reduces software management of the connection to almost nothing, and service routines to five instructions per word. Further, messages are not buffered in the satellites and this saves message copying times (and buffer space). At present the slight improvement we would gain with DMA connections would not justify the greatly increased costs and complexity involved.

It was clear we wanted to use 16 bit transfers, but the cable related costs of parallel connections were prohibitive. Thus, we opted for 16 bit serial synchronous communications, which required only 10 twisted pairs. Synchronous communications avoid many problems, and anyway, fast 16 bit asynchronous UART's were not available at the time.

Satellites require a ROM for bootstrapping, and we decided to integrate this into the design as well. The on-board ROM also has basic maintainence tests for the board, which can isolate a fault down the board or cable involved. This maintainence feature is most useful on satellites that dont have any other device capable of loading a maintainence program.

The resulting board has full duplex, 16 bit serial synchronous communications, with on-board opto-couplers, line-drivers and 256 word ROM. The vector and address are changed by DIP switches. The boards can be jumpered to operate between 300k to 1.2M baud, over distances of upto 200 meters. Separate Q-BUS and UNIBUS boards were developed.

The boards performance has satisfied our expectations; they are very fast, highly reliable and economic. The boards are as easy to install as a console interface and have presented no problems.

- Software considerations

The quality of intra-computer connection has a direct impact on the software that must supervise it. Unreliable connections require error correcting software, and this software occupies a lot of memory and takes a lot of time. The further step of implementing this supervisory software in hardware is both complex and expensive.

High speed and error free hardware connections simplify the work that has to be done in supervising an intra-computer connection. The devices described here operate with the reliability usually associated with disks and the supervisory software requires only error detection capability. A transmission error signals either a hardware failure on the board or cable, or a software problem (i.e. data being modified during transmission).

- Summary

Fast, reliable, low cost intra-computer connections across short distances, on private lines greatly simplify the task of building networks. We conclude this paper with a quotation from the DDCMP (DECnet) specifications:

"In the design of computer communications networks one of the basic considerations is the physical transmission of data from one computer to another. In the absence of of transmission errors this becomes a relatively simple task." (DDCMP specifications, edition 3, 1974)

- Acknowledgement

We would like to thank the Department of Neurobiology at the Max Planck Institute for Biophysical Chemistry, in Goettingen, where much of the early experience with intra-computer connections was first gained.

SAN DIEGO SYMPOSIA

THE FALL DECUS SYMPOSIA, SAN DIEGO, NOVEMBER 4-7
by Ray Strackbein, RT-11 Symposia Coordinator

I've just got to take this opportunity to give a little pep-talk to our readers about the FALL 1980 DECUS SYMPOSIUM.

For our readers who have never been to a symposium before, I must point out what an incredible experience a Decus Symposia can be. This will be my eighth symposia. At my first symposia I acquired a free "Documentation Directory" which listed every publication available from DEC and from it I was able to order necessary reference material which I hadn't known existed; I attended a session led by a member of the RT-11 development group who explained about their plans for the then-planned future release of Version 3. This allowed me to concentrate my efforts in more productive ways by planning my programs to anticipate some of the future features in RT-11; I overheard a couple of users complain about a part of their system which didn't work properly and I verified this with a few other users and delayed my plans for its use until I was able to verify it worked at a later date. The time and money spent on the Symposium was repaid in much, much higher productivity directly as a result of that first Symposium.

Not only is the symposium valuable for the user, but it is valuable for DEC as well (otherwise I am sure it wouldn't receive the DEC commitment it does). In my opinion, the major benefit which DEC receives from a Decus Symposium is the knowledge which DEC receives from its users. DEC's management and department heads listen very closely to everything the users say at a Symposium. From this they learn what they are doing wrong as well as what they are doing correctly. By listening to the users throughout the country they are able to monitor the performance of their regional offices. I once received a personal apology form the manager of the Los Angeles office because I told the people in one of the DEC booths that I was unhappy because my calls weren't being returned when I called the Los Angeles office.

Members of each of the DEC product lines will be at this Symposium and I assure you that they are very eager to hear about your experiences with the product for which they are responsible.

This is especially true of the RT-11 development group. They want to hear about just how you are using RT-11 and why you are using it (instead of DOS, CAPS, or RSX). They want to know what features you use most in RT-11 as well as those features you use the least and those features you need which RT-11 doesn't provide.

Now let me selfishly plug my pet project for this symposium.

Thursday night we are offerring a new session: RT-11 CAN DO IT (BELIEVE IT OR NOT) WORKSHOP. This session is intended to be a workshop on using RT-11 which will be entirely user-driven. You are encouraged to bring your problems to this workshop for us all to discuss. Just as welcome are explanations about personal short-cuts and programming tricks using RT-11. This session will run as late into the night as is necessary; however if it runs too long we may need some volunteers to get the beer and pizza.

It is my hope that this session will be of value to those fabulous people who gave us Version 4 (as they scheme up Version 5) as well as those of us who use it. I really hope we can make this session fun, intensive, and informative.

To that end I have recruited some incredible people to take the microphone and ham it up in an effort to cajole every pertinent idea out of the audience. In alphabetical order: Anton Chernoff (of some Massachusetts computer company), Greg Adams (employed by a foreign government), Nick Bourgeois (who works for Uncle Sam -- equal time), and Ray Strackbein (who works as little as possible).

Also scheduled for this symposium is a short talk by Bill Hubbard who will explain a memory protect schema for the PDP-11/34; a longer talk by Phil Sherrod who will explain the TSX-Plus time sharing system for the licensed RT-11 user; Mr. A. Hermes III and Mr. R. C. Strackbein will lead an assembly language tutorial which will concentrate on explaining the philosophy of the Unibus; Judy Hall is again sponsoring both the RT-11 Product Panel which gives an overview of RT-11 and its associated products and services as well as the RT-11 Feedback Session during which the Symposium Suggestion Box is opened and responded to; Judy also snuck (sneaked?) in an RT-11 Layered Products Panel which will give quick overviews about the various languages (Fortran and Basic (MU and SU) as well as Macro (What??? -- no APL?)) and enhancements to the vanilla (well, maybe chocolate-mint) RT-11 system (CTS-300, FMS-11, DEC-net, and Lab products); Jim Crapuchettes will talk about his SD handler which will let a device also become a sub-device (one way of reducing naming conflicts -- make a big file of several little files); Christopher Kelly will talk about an RT-11 CAMAC network using Fortran callable subroutines; and of course, John Rasted gets the prize for again leading three RT-11 sessions -- the Roadmap (this paragraph expanded to an hour and a half), the Sig Business Meeting where we talk about the past and future of the RT-11 Sig, and the Sig Wrap-up session where we flog the dying symposium in an effort to make the next one even better.

If you hadn't thought about attending the San Diego Symposia, please think about it. Thirty-five hundred DEC users can't be wrong.

P. S. If you are planning to attend, apply for an absentee ballot -- the symposium starts on election day!

SPR'S

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

PROBLEM
STATEMENT

Incorrect inline code is generated for the expression

```
        J=MIN0(I,10)/2
```

RESPONSE

The problem with the compiler will be fixed in the next rel
the meantime, please use threaded code as a temporary workar

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

PROBLEM
STATEMENT

Incorrect inline code is generated for assigning LOGIC
LOGICAL*1 variables.

RESPONSE

The problem with the compiler will be fixed in the next rele
the meantime please compile using threaded code as a workaround.

```
SOFTWARE  SYSTEM(S)  AFFECTED:  RT-11            VERSION(S): V03B-00

SOFTWARE COMPONENT(S) AFFECTED: Monitor          VERSION(S): V03B-00J
```

STATEMENT: Programs using .SCCA may still be aborted with a single Control C.

RESPONSE: Thank you for bringing this to our attention. This is a restriction in V03B (unfortunately undocumented). The input to the terminal itself does not intercept Control C if .SCCA is in effect. This is true anytime TT: is used as an input device. This restriction will be published in the RT-11 V04 Documentation set.

```
OFTWARE SYSTEM(S)  AFFECTED:  RT-11           VERSION(S): V03B-00

OFTWARE COMPONENT(S) AFFECTED: System User's  VERSION(S): DEC-11-ORGDA-A-D,
                               Guide                      DN1
```

TATEMENT: LINK/RUN/INCLUDE in files generates the error message

```
        ?LINK-W-Illegal character
```

SPONSE: As it turns out, you cannot use /RUN with any option that requires a response from the terminal. This is an undocumented restriction. Thank you for bringing this to our attention.

In your case, when RT-11 parsed your monitor command,

```
        LINK/RUN/EXE:SY:B BANG,SY:RTLLIB/INCL
```

RT-11 created internally an indirect command file that looked like this:

```
        .RUN SY:LINK
        *SY:B.SAV=DK:BANG.OBJ,SY:RTLLIB.OBJ/I
        *^C
        .RUN SY:B.SAV
```

When the linker looked for the library files indicated by the /INCLUDE option, it read the line

```
        .RUN SY:B.SAV
```

The colon in SY:B.SAV caused RT-11 to generate the error message you received.

You can get around this restriction by using an indirect command file as follows:

```
.LINK/RUN/EXE:SY:B BANG,SY:RTLLIB/I
ROO
<RET>
```

This restriction will be documented in a future release of RT-11.

```
SOFTWARE  SYSTEM(S)  AFFECTED:  RT-11            VERSION(S): V03B-00

SOFTWARE COMPONENT(S) AFFECTED: SYSTEM USER'S    VERSION(S):
                                GUIDE            DEC-11-ORGDA-A-D,DN1
```

STATEMENT: For magtape, the INITIALIZE/VOLUMEID:ONLY command initializes the entire volume as well as the volume identification. This is not documented.

RESPONSE: You are correct. This restriction is documented in the next release of RT-11. We regret any inconvience this restriction has caused you.

RT-11 MARKETPLACE

Dear Mr. Demers:

Enclosed is a description of an expanded cross-reference utility which I have developed. I would appreciate it if you would include it in the next RT-11 MiniTasker.

I developed the PREF/PMAT utility, a system-level, cross-reference utility, because the use-envelope implied/generated by the DEC CREF utility is so limited. CREF isn't particularly valuable for unit development as the programmer really is on top of what is going on in his unit. At the system integration level, CREF is so cumbersome that most projects develop a manual "interface" document. I use the system-level, cross-reference document in system integration and system maintenance/expansion. In addition to easing the unit-document search problem, the PREF/PMAT document also eases the pain when LINK reports that a global symbol is undefined. One lookup in the system-level document will show where that symbol is erroneously defined as a local symbol.

Further information about this product can be obtained from me at the above address.

Hollis D. Paul
Lilliputian Systems, Inc.
5112 88th Street S.W.
Everett, WA 98204

Yours very truly,

Hollis D Paul

LILLIPUTIAN SYSTEMS, INC.
5112 88TH STREET S.W.
EVERETT, WA 98204

PREF
PMAT

System-Level Cross-Reference Utility

PREF/PMAT: A utility that allows the user to construct a single document which provides a cross-referenced listing of the occurrence of the symbols in all the modules of a software system.
PREF: A modified version of the CREF program which captures the per-module symbol table and cross-reference text in a separate, permanent file.
PMAT: A program which accesses the per-module cross-reference text files and formats the combined, system-level, cross-reference document.

FEATURES

- Runs under RT-11 V3B.

- Handles all symbol-type cross-reference tables which CREF produces on a per-module basis. (S,R,M,P,C,E; Default is /S/M/E)

- Allows the inclusion of the per-module symbol table.

- Orders the per-module cross-reference text for a symbol alphabetically by module title.

- Provides module name and symbol type (R,E,G,X) at the beginning of the first line of per-module cross-reference text.

- Handles up to 39 modules in a single system (depending upon system configuration).

- Uses SOB, MUL, and DIV instructions for program efficiency.

DESCRIPTION

PREF is a program which was created by inserting new functions into D.E.C.'s RT-11 supplied CREF utility (V01.08). These functional additions capture the symbol table from the MACRO-generated listing file, close the listing file, generate new entries for the CREF.TMP file which are used to pass the symbol type data to the sort/text generation code, and open the .FRC file to receive the cross-reference text. The original CREF sorting mechanism is unchanged. The text generation code is modified to prefix the module name, to process the new 0-line entry which contains the symbol type data, and to indent non-initial lines so that data entries are justified at field 1 (rather than field 0).

PMAT is a program which opens all the per-module cross-reference listing files generated by PREF and interleaves the cross-reference text. The input/output requirements are given to PMAT in an indirect command file. (PMAT does not do directory searches for all .FRC files.) The user specifies the output file name and the names of input files on succeeding lines. PMAT then opens the input files, processes the symbol tables, and interleaves the cross-reference text for output.

The user can specify the following options to control the generated output. Any of these options may be preceeded by NO to suppress the option.

/TABLE	- Include the full symbol table with the title in the document header.
/UPDATE	- Display program progress messages on the control console.
/SYM	- Output a User Symbol Cross-reference Table.
/REG	- Output a Register Symbol Cross-reference Table.
/MACRO	- Output a Macro Symbol Cross-reference Table.
/PERM	- Output a Permanent Symbol Cross-reference Table.
/CONTROL	- Output a Linker Control Symbol Cross-reference Table.
/ERROR	- Output the Error Symbol Cross-reference Table

The default options are /TABLE/UPDATE/SYM/MACRO/ERROR. The user cannot successfully request a cross-reference table of a symbol type that was not requested when the individual modules were assembled.

USE ENVELOPE

The PREF/PMAT utility was designed to be used as a tool in three specific functions of the over-all software development task:

- System Integration and Maintenance - The interleaved cross-reference text provides an easy means of determining
  1) if a symbol is globalized in all modules,
  2) the module in which a global symbol is defined, and
  3) the modules which modify the value of a variable.

- Design Verification - The collection of the system-wide symbol definitions in a single document by a utility facilitates
  1) the complete, automatic reporting of symbols actually defined in a system,
  2) the monitoring of the use of symbol naming conventions, and
  3) the elimination of the multiple, local definitions of a variable of global significance.

- System Documentation - The military programming standards (specifically, the avy WS-8506, but also the similar Army and Air Force standards) require cross-eference tables and referenced-modified-initialized matrices for modules and data-base items. The PREF/PMAT utility provides all of this data, and clever use of program design and naming conventions can break out the categories acceptably.

```
PMAT      OPSHUN   0-RG       3-17#     6-97*     7-10      8-30      9-43*
                   14-5      14-19     14-54     14-89     15-13
PMATSO    OPSHUN   0-=G       1-11      5-17      5-36      7-169


PMAT      WRILIN   0-=G       1-15     14-39     14-44     14-49     14-57
PMATSO    WRILIN   0-=G       1-6       7-51
PMATWR    WRILIN   0-RG       6-13#    10-15

PMAT      WRINIT   0-=G       1-15      5-21
PMATWR    WRINIT   0-RG       3-4#

PMAT      WRIPAG   0-=G       1-15
PMATSO    WRIPAG   0-=G       1-6
PMATWR    WRIPAG   0-RG      10-9#     10-22     10-25
```

NEW PRODUCT ANNOUNCEMENT - ZORK^{tm}

"ZORK: The Great Underground Empire - Part I", a fantasy simulation game previously available only to users of large computers, is now available for the DEC PDP-11. This version of the game was developed by the original authors based on their ZORK (Dungeon) game for the PDP-10. It features a greatly improved parser; command input and transcript output files; SAVEs to any device and file name; and adaptation to different terminal types, including a status line on VT100s. Note: this is not the FORTRAN translation that has been available through DECUS. This version has been completely rewritten to run efficiently on small machines - up to 10 times as fast as the DECUS version.

ZORK has a vocabulary of over 600 nouns, verbs, adjectives, prepositions, pronouns, and articles, allowing a player to converse with it in a rich dialog, much like natural English. For example, ZORK would understand the sentence, "Put all the books but the red one in the bag." Imagination-provoking prose descriptions of the diverse regions of the Great Underground Empire are revealed on the screen to the player as he or she explores over 100 rooms, containing more than 150 objects, and solves some 25 intricate, ingenious, and often diabolical problems.

This ZORK game, the first in a series, is for the explorer who enjoys a challenge, and it will provide dozens or even hundreds of hours of fun and recreation. ZORK runs under RT-11, HT-11, or RSTS/E and requires as little as 20K words of memory and a single floppy disk drive. The game package, consisting of an RX01-format diskette and an instruction booklet, is available from Infocom, Inc., P.O. Box 120 Kendall Station, Cambridge, Ma. 02142

A R E C - ACCOUNTS RECEIVABLE SYSTEM FOR RT-11

AREC is a highly interactive accounts receivable package developed to run on a small RT-11 based system. It is a flexible and easy-to-use package which may be implemented as either an open item or balance forward accounts receivable system. AREC tracks all or current (in a balance forward implementation) invoices, credit memos, cash receipts and adjustments, and will produce monthly statements for all or selected customer accounts. The system also produces sales analysis and other additional management reports on an "as needed" basis. An inquiry facility allows the user to display the current status of any customer account.

The minimum configuration is a PDT-150 with dual floppies, 56K bytes of memory and a printer. AREC is written in FORTRAN IV and requires RT-11 V3B or later.

For further information contact:

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

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.