THE mini-tasker

DECUS

RT-11 SIG NEWSLETTER

AUGUST 1975 VOL. 1 NO. 3

FROM THE EDITOR

I must apologize for the delay in the publication of this issue of the newsletter. Now that most of the summer vacations are a thing of the past, I am looking forward to an increase in the number of contributions to the Mini-Tasker. Any comments or suggestions in regard to the format of the newsletter would be most welcome.

There is a problem with reproducing the articles submitted for the newsletter. The DECUS office cannot produce the newsletter directly from xerox copies. I am, therefore, seriously considering using RUNOFF as an aid in producing the Mini-Tasker. I would be interested in hearing from anyone who might have a copy of RUNOFF or any other software that would be helpful in this endeavor. I would be running it on a 28K processor under RT-11.

I would like to thank those who have contributed to this issue of the newsletter, and hope they will continue to do so. It is my intention ot produce the Mini-Tasker on a monthly basis, but to do this I will need contributions from additional sources. Lets hear from some of the other committees that were formed at the Spring DECUS meeting.

Please send completed items to me at:

John T. Rasted
CAM Systems, Inc.
17 Brown Street
Waterbury, CT 06702
(203) 757-8010

News that is to be compiled into articles should be sent to:

Seldon E. Ball, Jr.
Wilson Lab. of Nuclear Studies
Cornell University
Ithaca, NY 14853
(607) 256-4882

OPEN DR. MEMORY

Being a test site for RT-11 and FORTRAN, I've been using Version 2B, FORTRAN, SYSLIB and BATCH since January. Here are a couple of tidbits I thought might be helpful to others as V2B becomes generally available.

SYSLIB is a great help since it provides the FORTRAN programmer convenient access to all RT-11 programmed requests. It does, however, put the FORTRAN programmer much closer to the system and allows him to fall into many holes usually found only by the assembly language programmer. USR swapping is a good example; since many of the SYSLIB functions dealing with files cause the USR to swap (if it is non-resident), one must be very careful to construct the program so that it does not swap over something, like a file name, it needs while swapped. One way to do this is to call the SYSLIB routine only from an overlay, since overlay areas are safely out of the swapping area. One must also be careful that the SYSLIB routine itself, which is usually near the top of the program (being a library routine) does not get swapped over. If the FORTRAN program uses a common block, it will normally be swapped over.

BATCH is likewise a great help, I've had no problems with it at all. I find it useful to set up BATCH streams to link each of my main programs, some of which have 8-1Ø lines of link command string input. Then I can relink them by running BATCH instead of typing the whole link command input each time. Another BATCH stream, which calls each of the individual BATCH streams, allows me to relink all the programs at once. This is a handy way to document the required link commands for each program, too.

The 'OLD' specification in system subroutine 'ASSIGN' is useless. In particular, it does not provide any file protection for old files, as is implied. If an existing direct access file is opened with a write, it will still be truncated upon closing to the last record written!!!

V2B introduces a 'CLOSE' subroutine which can be used to close a file, thus freeing its buffer space for reuse. Since 28K can quickly get crowded with a foreground and a background program, each of which uses several files, I use this a lot. However--one must be very careful not to call 'CLOSE' for a file which is not open. If this happens, 'CLOSE' opens the file and then immediately closes it. This results in your original file being deleted and replaced by a new file of Ø length. This may be a handy way to delete files, if you don't mind cluttering up your directory with Ø length files, but it is also a good way to delete an important file. The FORTRAN 'END FILE' statement now writes an end-of-file record into the file but does not close it or de-allocate the buffer. If 'CALL ASSIGN' was used to establish other-then-default file names, 'CLOSE' replaces the assigned name with the default, so that if you need to reopen the same file later, you must redo the 'ASSIGN'.

I get around both these problems by always explicitly opening existing files with a 'FIND' immediately upon starting a program or subroutine. This opens the file with a 'LOOKUP' and the existing file is thus safe.

FORTRAN limits you to 15 channels and to 15 files that can be 'ASSIGNED' at any one time (using the /N:17 switch). The number of channels cannot be increased without SYSLIB, but the number of files that can be 'ASSIGNED' can be increased by linking the following routine before the FORTRAN mainline:

```
        .TITLE CHANEL
        .GLOBL $NLCHN
        $NLCHN=??.
        .END
```

Where ?? is the number of files you want to be able to assign. Each one costs you about 17 words of core. Then link like this:

```
        *MAIN,LP:=CHANEL,MAIN/F          ETC.
```

The error message "MULTIPLE DEFINITION OF $NLCHN" will appear twice during the link but can be ignored. Remember, only 15 files can be open at any one time.

Stack overflow is something to watch for, too. Since FORTRAN makes heavy use of the stack, particularly for subroutine calls, and context switching takes 2Ø-odd words. If the system context switches while nested several deep in subroutines, the stack can overflow. The 'FRUN' command allows only 128 (1Ø) words for the stack and one of my programs overflows this if it trys to context switch while in a 3rd-level subroutine. This problem is hard to find because no error message is issued and the foreground program just 'GOES AWAY' without locking out the background. Use the 'FRUN' /S switch to allocate some extra stack space if funny things happen to your foreground programs.

I find a simple spooling program that can be put in the foreground very useful. It wakes up every few seconds and looks for any files with a '.LST' extension. If it finds any, it lists them on the line printer before going to sleep again. This speeds up program development greatly, since all listings can be directed to disk. Such a spooler is listed in the new RT-11 software support manual. I'll put a fancier version with some switches into the DECUS Library soon.

I've had no problems at all using FORTRAN and SYSLIB in the foreground (except things get crowded fast). You can even allow a foreground and a background program to access the same files at the same time.

I'll be glad to try to help any user that has FORTRAN problems. Give ma a call at (513) 426-3111.

Good luck! You'll like V2B.

DAVE SYKES

COMMENTS FROM DEC

Bob Bean's Directory Extension program which was inadvertently omitted from inclusion in the last issue of the Mini-Tasker is included below.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;DIREXT-PROGRAM TO ADD SEGMENTS TO AN RT-11 DIRECTORY
;TO USE,THE FIRST ENTRY IN THE DIRECTORY MUST BE <UNUSED>,AND ITS LENGTH
;MUST BE TWICE AS MANY BLOCKS AS THE NUMBER OF SEGMENTS THAT ARE TO
;BE ADDED. THE FIRST ENTRY MAY BE MADE <UNUSED> SIMPLY BY CALLING
;PIP TO MOVE THE FIRST FILE ON THE DEVICE. ONCE THE UNUSED AREA HAS BEEN
;CREATED,DIREXT IS RUN BY TYPING
;                .R DIREXT
;THE RESPONSE:   *
;A COMMAND LINE IS ENTERED OF THE FORM "DEV:/N:M" WHERE DEV IS
;THE DEVICE WHOSE DIRECTORY IS TO BE EXPANDED,AND "M" IS
;THE NUMBER OF SEGMENTS (NOT BLOCKS!) TO ADD TO THE DIRECTORY.
;FOR EXAMPLE:
;                *RK:/N:14       WILL ADD 12 (DECIMAL) SEGMENTS TO THE
;                                DIRECTORY ON RKØ, FOR WHICH THE FIRST
;                                FILE MUST BE AN <UNUSED> AREA OF AT
;                                LEAST 24 BLOCKS
;
;THERE ARE TWO POSSIBLE ERROR MESSAGES.
;       ?HARD I/O ERROR?         WILL OCCUR IF THE DIRECTORY FOR THE
;                                DEVICE CANNOT BE READ OR WRITTEN WITHOUT
;                                HARDWARE ERROR.
;       ?ILLEGAL COMMAND?        WILL OCCUR ON ANY OF THE FOLLOWING CONDITIONS
;                                1)-INAPPROPRIATE COMMAND STRING
;                                2)-FIRST ENTRY IN DIRECTORY NOT <UNUSED>
;                                3)-REQUESTED NUMBER OF SEGMENTS WOULD
;                                   CAUSE TOTAL TO EXCEED ALLOWABLE 31(1Ø)
;                                4)-SIZE OF <UNUSED> AREA TOO SMALL TO
;                                   ACCOMODATE REQUESTED DIRECTORY INCREASE
;IF THE ?ILLEGAL COMMAND? MESSAGE OCCURS,THE DIRECTORY IS UNMODIFIED,
;AND THE PROGRAM IS RESTARTED.
;
;TO CREATE DIREXT,TYPE THIS TEXT INTO THE FILE "DIREXT.MAC" WITH THE EDITOR.
;TO ASSEMBLE,    .R MACRO
;               *DIREXT=DIREXT
;TO LINK,        .R LINK
;               *DIREXT=DIREXT
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
        .MCALL  ..V1..,.CSIGEN,.READW,.WRITW,.EXIT,.PRINT
        ..V1..
        R0=%0
        R1=%1
        R2=%2
        R3=%3
        R4=%4
        R5=%5
        SP=%6

ILLCMD: .PRINT  #MSG1           ;TYPE "?ILLEGAL COMMAND?"
START:  .CSIGEN #DSPACE,#DEXT,#0        ;GET COMMAND STRING
        DEC     (SP)+           ;ANY SWITCHES?
        BNE     ILLCMD          ;NO-BAD COMMAND
        CMP     (SP)+,#101516   ;/N:M ON FIRST INPUT FILE?
        BNE     ILLCMD          ;NO-BAD COMMAND
        MOV     (SP)+,R1        ;NUMBER OF SEGS TO ADD INTO R1
        MOV     #6,R2           ;BLOCK FOR CURRENT SEG IN R2
        MOV     #DIRBUF,R3      ;R3 POINTS INTO DIRECTORY BUFFER
        .READW  3,R3,#256.,R2   ;READ FIRST DIRECTORY SEG
        BCS     HERR            ;READ ERROR
        CMP     12(R3),#1000    ;FIRST ENTRY <UNUSED>?
        BNE     ILLCMD          ;NO-BAD COMMAND
        MOV     (R3),R4         ;NUMBER OF SEGS NOW IN DIR INTO R4
        ADD     R1,R4           ;TOTAL SEGS FOR NEW DIR IN R4
        MOV     R4,R5           ;REMEMBER NEW SEG TOTAL
        CMP     R4,#37          ;IS NEW TOTAL TOO LARGE?
        BHI     ILLCMD          ;YES-BAD COMMAND
        ASL     R1              ;# OF BLOCKS NEEDED FOR NEW SEGS
        CMP     R1,22(R3)       ;<UNUSED> LARGE ENOUGH TO TAKE NEW SEGS?
        BGT     ILLCMD          ;NO-BAD COMMAND
        ADD     R1,10(R3)       ;UPDATE FILE START ADDRESS
        SUB     R1,22(R3)       ;REDUCE <UNUSED> SIZE
LOOP:   MOV     R4,(R3)         ;UPDATE NUMBER OF SEGS TO NEW TOTAL
        .WRITW  3,R3,#256.,R2   ;WRITE SEG BACK OUT
        BCS     HERR            ;WRITE ERROR
        DEC     R5              ;ALL SEGS UPDATED?
        BLE     DONE            ;YES
        TST     (R2)+           ;NO-ADD 2 TO BLOCK NUMBER
        .READW  3,R3,#256.,R2   ;READ NEXT SEG
        BCC     LOOP            ;AND UPDATE
HERR:   .PRINT  #MSG2           ;PRINT "?HARD I/O ERROR?"
DONE:   .EXIT
MSG1:   .ASCIZ  "?ILLEGAL COMMAND?"
MSG2:   .ASCIZ  "?HARD I/O ERROR?"
        .EVEN
DEXT:   0,0,0,0
DIRBUF: .=.+1000
DSPACE:
        .END    START
```

LUGS

The following individuals are interested in forming local user groups in their respective areas.

David Sykes
Mead Technology Laboratories
3481 Dayton-Xenia Road
Dayton, OH 45432
(513) 426-3111

Gerry Forshee
Systems Analyst
Indiana University
Bloomington, Ind. 47401
(812) 337-6122 office
337-1768
339-0950 home

INSTALLATIONS

Mead Tech Labs uses RT-11 to run a rather extensive management information, scheduling, and scientific data collection and reduction system. We have two CPUs, the larger of which is connected to a Diano Hardy Spectrophotometer; the smaller will soon be going in a camper-type van to control and collect data from radiometers in the field through a LPS. We also have a special interface to a 'STR-LINK' digital cassette recorder made by Electronic Processor Inc. All programs are in FORTRAN.

System 1:

PDP-11/2Ø
28K of core
TU1Ø 7-track tape drive
TU6Ø DECassette drive
3 RKØ5 disk drives
VTØ5
Hardy Spectrophotometer interface
STR-LINK interface
Versatec 11ØØA printer/plotter
Teletype

System 2:

PDP-11/1Ø
16K of core
DECwriter II
TU6Ø DECassette
RKØ5
LPS (soon)

I'd be glad to hear from anybody in the Dayton area interested in a lug. I'm the SIG Library Chairman as well.

David Sykes
Mead Technology Laboratories
3481 Dayton-Xenia Road
Dayton, OH 45432
(513) 426-3111

FOCAL/RT-11

A presentation of FOCAL-11 was made at Spring DECUS by Steve Mullen from Life Science Marketing in the Laboratory Data Products Group of Digital Equipment Corporation. There are two enhanced versions of FOCAL-11 currently available. One version, FOCAL/PTS, is designed for non-mass storage systems, while the second release, FOCAL/RT-11, operates under RT-11. The latter version has extensive library support such as program and data storage, as well as virtual files.

Also discussed was a major real-time extension to FOCAL/RT-11 which will be released in the near future. This extension provides very sophisticated support for experiment control and data acquisition. Although oriented toward the life sciences, it should be fully applicable to all laboratory science research. Extensive information on the full FOCAL-11 system will be soon published in the spring DECUS proceedings.

SPRS

digital **SOFTWARE PERFORMANCE REPORT**

FIELD #: | SPR #: FOR DEC USE ONLY

11560

Page ____ of ____

SYSTEM PROGRAM AND VERSION (OR DOCUMENT)	MONITOR AND VERSION	DATE
FORTRAN	RT-11 V02-01	26-MAR-75

NAME: Mark Bartelt
FIRM: California Institute of Technology, MS 356-48
ADDRESS: Pasadena, California 91125
ZIP

DEC OFFICE

SUBMITTED BY: Mark Bartelt PHONE: 795-6811 X2663

LIST ATTACHMENTS

REPORT TYPE
- ☐ LOGIC/CODING ERROR
- ☐ DOCUMENTATION ERROR
- ☒ SUGGESTION
- ☒ INQUIRY
- ☐ FOR YOUR INFORMATION

PRIORITY
- ☐ LOW
- ☐ STANDARD
- ☒ HIGH

CAN THE PROBLEM BE REPRODUCED AT WILL?
☐ YES ☐ NO

CPU TYPE	SERIAL NO.	SYSTEM DEVICE	MEMORY SIZE	DISTRIBUTION MEDIUM
PDP-11/45		RK05	16K	RK05

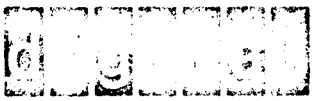
Problem:

RT-11 FORTRAN I/O package is not reentrant. It should be. Why isn't it? Will it ever be? If so, when?

This is a serious deficiency. For example, suppose a job issues a mark-time request. The mark-time completion routine saves registers, and invokes a FORTRAN subroutine which issues FORTRAN I/O calls. The program dies (via error 26) if, at the time that the mark-time completion routine gains control, the job which issued the mark-time request is itself in the middle of a FORTRAN I/O call.

Of course, one can kludge around by setting and resetting a global flag before and after all FORTRAN I/O statements, and forcing the mark-time completion routine to interrogate the flag and, if set, release control for a few clock ticks by issuing a new mark-time request and returning without doing anything else. But who wants to mess around with this sort of clumsiness?

Please give us a reentrant FORTRAN I/O package soon! Or, as a consolation prize, perhaps you could suggest a less inelegant way of dealing with this problem.



14 April 1975

Mr. Mark Bartelt
California Institute of Technology
MS 356-48
Pasadena, CA 91125

The following is in response to SPR #31-4602 concerning RT-11 V1-11.

PROBLEM:

FORTRAN I/O package is not re-entrant.

DISPOSITION:

Your suggestion will be seriously considered. We are aware that this would be a very useful feature. At present we are collecting ideas for a future version of FORTRAN. You can feel confident that your suggestion will be at the top of the priority list.

For inquiries concerning the above, please contact:

Software Communications
P. O. Box F
Maynard, MA 01754

dmh

digital

SOFTWARE PERFORMANCE REPORT

FIELD #: | SPR #:
FOR DEC USE ONLY

16983

Page ____ of ____

SYSTEM PROGRAM AND VERSION (OR DOCUMENT)	MONITOR AND VERSION	DATE
FORTRAN	RT-11 V02-01	16-JUL-75

NAME: Mark Bartelt
FIRM: California Institute of Technology, MS 356-48
ADDRESS: Pasadena, California 91125
ZIP

SUBMITTED BY: PHONE:

LIST ATTACHMENTS

DEC OFFICE

REPORT TYPE
- [X] LOGIC/CODING ERROR
- [ ] DOCUMENTATION ERROR
- [ ] SUGGESTION
- [ ] INQUIRY
- [ ] FOR YOUR INFORMATION

PRIORITY
- [ ] LOW
- [ ] STANDARD
- [X] HIGH

CAN THE PROBLEM BE REPRODUCED AT WILL?
[X] YES [ ] NO

CPU TYPE	SERIAL NO.	SYSTEM DEVICE	MEMORY SIZE	DISTRIBUTION MEDIUM
11/45,11/10		RK05	16K;28K	RK05

The enclosed listing illustrates a compiler bug which has been troubling us. My guess is that the optimizer is trying to be a bit too clever, and ends up outsmarting itself.

As the listing shows, the problem seems to be connectec somehow with the fact that the array which has one of its elements incremented also appears in an 'EQUIVALENCE' statement, since the problem goes away when the 'EQUIVALANCE' statement is removed.

```
.R PIP
*TT:=BUG.FOR
        IMPLICIT INTEGER (A-Z)
        DIMENSION A(3,5),X(3),Q(3,20)
        EQUIVALENCE (Q(1,10),X(1))
        DATA S/2/
C
        X(S)=0
C
        DO 1 I=1,5
        X(S)=X(S)+1
        A(S,I)=X(S)
1       CONTINUE
C
        TYPE 2, (A(S,I),I=1,5)
2       FORMAT (5I3)
C
        CALL EXIT
C
        END
*^C

.R FORTRA
*BUG=BUG
*^C

.R LINK
*BUG=BUG/F


*^C

.R BUG

 0  1  2  3  4
.
```

```
.R PIP
*TT:=NOBUG.FOR
        IMPLICIT INTEGER (A-Z)
        DIMENSION A(3,5),X(3),Q(3,20)
        DATA S/2/
C
        X(S)=0
C
        DO 1 I=1,5
        X(S)=X(S)+1
        A(S,I)=X(S)
1       CONTINUE
C
        TYPE 2, (A(S,I),I=1,5)
2       FORMAT (5I3)
C
        CALL EXIT
C
        END
*^C

.R FORTRA
*NOBUG=NOBUG
*^C

.R LINK
*NOBUG=NOBUG/F


*^C

.R NOBUG

 1  2  3  4  5
.
```

DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
MAYNARD, MASSACHUSETTS 01754

ADDRESS CORRECTION REQUESTED

BULK RATE
U.S. POSTAGE
PAID
DIGITAL EQUIPMENT
CORPORATION