DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER

September Number 18 1976

Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator
12 Bit SIG
c/o DECUS .. OR .. Liberty Mutual Research Center
146 Main Street 71 Frankland Road
Maynard, MA 01754 Hopkinton, MA 01748

(Please include reference to Newsletter by number and page when inquiring about material publisned.)

DECUS/Europe contributions are solicited by:

Lars Palmer
DECUS/Europe 12 Bit SIG Newsletter Liaison
Hassle
Fack
S-431 20 MOLNDAL 1
SWEDEN

NEWSLETTER DEADLINE

Deadline for ready-to-use material for the next Newsletter is October 29. Material requiring editing/re-typing must be in earlier. Note: a review of material in recent Newsletters will show what sort of submissions reproduce well and what do not.

SIG BUSINESS

To date no formal response has been received to our request to the DECUS/US Executive Board for representation of our 12 Bit Special Interest Group on the Board. The request was presented in May. Informal communications indicate that the Board intends to delay any action on the application at least until the December meeting in Las Vegas. So far no further word is available on the subject. I expect that we will discuss it in the SIG meeting at the Symposium in December. In any case we need interested members with ideas and energy to contribute to the activities of our Special Interest Group. In particular work on the Symposium and contributions to the Symposium are needed, and we would like to build up local user groups and hopefully regional get-togethers of 12 Bit users on a comparatively informal, low-cost basis for exchange of ideas and trading help and maybe even actually writing programs that are needed by the user community. We also need inputs on other new and creative ideas for SIG activities.

FALL DECUS SYMPOSIUM

The schedule for the Fall Symposium to be held the week of December 6 at the MGM Grand Hotel in Las Vegas has been set. It contains an interesting and more extensive program for 12 Bit users than we have had the last few meetings There will be the usual Product Panel for the PDP-8. We will have a business meeting for the Special Interest Group where several important matters will be discussed and inputs from users will be solicited, and there will be a workshop on OS/8 topics all scheduled for the first day. Notice that the first day is Monday this time. The symposium is running from Monday through Thursday rather than the normal Tuesday through Friday. On the second day there are plans for workshops on the FPP-8a floating point processor, on MACREL and its LINKER and on RTS-8. Also on Tuesday there will be a DECnet-8 session. On Wednesday afternoon there is a session of OS/8 application papers. Also scattered throughout the program are other items of interest to the 12 Bit community and although the program schedule is quite tight we are planning to leave provisions in the scheduling to try to fit in late developing material. One of the ways that we will be handling this is through birds-of-a-feather sessions which can be set up almost on the spot.

Due to the incredible backlog that has developed in PDP-8 products lately no one knows what hardware DEC will be able to send to the meeting yet.

DR. DOBB'S

I heard recently from Jim Warren of the People's Computer Company in Menlo Park, California, who is the editor of Dr. Dobb's Journal of Computer Calisthenics and Orthodontia ("A reference journal for users of home computers"). As many of you already know, "Dr. Dobb's" is a very interesting member of the growing community of publications aimed at micro-computer users, and in particular the "personal computing" segment. It concentrates more on the software side than some of the other publications such as BYTE. Jim wrote to me because he has his own PDP-8 and is interested in OS/8. I suspect that the personal computer world will soon start to discover OS/8 and the other PDP-8 software, first because it makes an excellent design model for a small, flexible operating system for any of the micro-computers but also because micro-computer class versions of the PDP-8 are now emerging and OS/8 will be the best software available for the more complete systems (i.e., ones with at least 8K and a floppy disk).

FOLLOW-UP ON DIRECT/A

Larry Fowler from Boeing wrote to remind me that the directory alphabetize work was originally his and it was distributed at the Fall 1975 DECUS Conference. You will recall that in a previous Newsletter I mentioned that his name had been lost in the scramble. Since mentioning in the previous Newsletter that a version of DIRECT with the /A option was available from me I have heard from Tom McIntyre that he now has available a further improved version which implements the alphabetize feature, a feature for printing DECsystem-8 labels and volume numbers if they are available, and a feature that I like very much of printing multiple column directories ordered vertically rather than horizontally which makes them much easier to use, I think. Because Tom's version is substantially superior to the previous ones, including mine, I am withdrawing my version in favor of Tom's which he has promised to submit to the DECUS Library very soon. In the meantime you should contact Tom if you have any questions regarding it. He is at the Department of Physiology and Biophysics at West Virginia University Medical Center in Morgantown, West Virginia 26506.

NOTES FROM JIM VAN ZEE

Jim has been in a flurry of activity in preparation for finishing up work at the University of Washington. In particular he has been adding even more features to U/W FOCAL and trying to make it into a good "final" form. It's hard for me to keep up with all of the enhancements that Jim keeps inventing. Among the latest is a function to access the OS/8 system date. He has added more "secret variables" and he now can once again support a 6 digit version of FOCAL although he still believes that the 10 digit version is by far the preferable version. Jim also has a way passed on to him by Steve Gillette of the U.S. Geological Survey in Flagstaff to add an "empties" option to the List commands so FOCAL can now print a complete map of a directory including empties with only the file dates missing. He also has invented an Open Second command for the 12 and 16K versions and, of course, an Open Restore Second command. This means that you can have two open files simultaneously. So now you can have two OS/8 files plus the terminal going at one time. Jim has had a series of corrections and improvements to his VC8E handler. The DECUS Library and I are sorting out the latest versions of it. It should be all taken care of and into the Library long before you receive this Newsletter. Jim has written a program to use many of the special functions of FOCAL to aid in reconstructing a crashed directory using a PIP /E listing of it. Similar things have been done before such as Tim Clark's version in TECO but this is a rather interesting demonstration of the capabilities of U/W FOCAL.

Enclosed elsewhere is an SPR from Jim regarding a problem that he ran into recently when he discovered that the latest release of PAL8 (Version 10) seems to have shrunk the symbol table rather substantially. People with large memory systems don't really know the difference, in fact, you don't have to use /K any more to use extended memory for your symbol table so you don't even know that this change has taken place, but if you have an 8K system that just barely assembled something like FOCAL before because of the limitation on the number of symbols you are now out of luck, apparently. Does anyone else have any input on this subject? In the meantime, I suggest that you be sure not to throw away your last copy of the old Version 9 if you have any possible need for the larger capacity.

MATERIAL FROM LARS PALMER

Lars has sent along a couple of paper tapes that incorporate:

1. The /M patch to OS/8 FORTRAN IV PASS3 that was mentioned in a previous Newsletter.

2. Several patches for FRTS:

    a. The input error patch as submitted by Jim Crapuchettes.

    b. The patches to FRTS for the USR subroutines by Bob Phelps.

    c. The patched FRTS allowing it to pick up command decoder switches. A subroutine using this patch will be submitted to DECUS soon.

    d. The core allocation patch to use BAT with FORTRAN IV.

NOTE FROM TIM CLARK

Tim has sent along word on a number of interesting developments. First, he has written and documented a TECO macro that will strip out comments from any TECO macro thus compressing it. This would be a very useful tool for everyone to use when they are writing complicated TECO macros as it allows them to fully document the macro but still conserve space at run time. The use of something like this seems almost mandatory if we are ever to establish a library of TECO macros for general exchange.

Tim also enclosed some information on a FXDIR technique for fixing directories using TECO and other OS/8 programs.

Tim reports on Jim Crapuchettes' latest work on FUTIL which will make several new features available. Final plans aren't set yet but he is trying to at least include support for FORTRAN IV load modules, the FPP instruction decoding, a BYTE mode for output and a WRITE with block number argument.

Other things of interest that Tim has done recently include:

1. Adding an instruction to OS/8 TECO that provides memory examine and deposit functions. This is useful for turning on and off echo and zero suppression and releasing space for two-page handlers. Of course, it has potential for causing no end of grief as he says.

2. TECO macro that, given an ASCII DIRECT file or any suitable list of file names, will generate a batch input file that will OCOMP on all the specified files on two devices. This is useful for comparing the results of a SQUASH transfer; for example, because OCOMP does an octal comparison between the files.

3. He has also worked up a set-up to allow the MR8E (TD8E ROM bootstrap) to boot either the TD8E system or another (typically a disk) system.

4. Finally, Tim is looking for a report on success with writing LINCtapes on a TD8E. He suggests someone with experience should write it up for the Newsletter.

Tim says that anyone wanting copies of CMPRS and FXDIR can send him two DECtapes or LINCtapes or floppy disks and he will send one of them back with the appropriate material on it. His address is: Frelan Associates, PO Box 298, Menlo Park, CA 94025.

NOTE FROM AREND KUMAP

Mr. Kumap writes to note that the available patch for TIME problems of FORTRAN IV does not work correctly. No further information was included. He notes that FLAP and the FPP support library are still available from DEC, they work, and he indicates that he is interested in learning more about RTS-8 and FPP-12 programs. He is ready to cooperate with others on the subject. He has a 16K PDP8/E with FPP-12 and RKØ5. His address is: Universiteit van Amsterdam, Laboratorium voor Psychofysiologie, Eerste Constantijn Huygensstraat 20, Amsterdam, The Netherlands.

NOTE FROM CHARLES RAY SMITH

Ray sent along information on something he calls PSD. It is a PS/8-OS/8 symbolic de-bugger. This is rather like the DDT that is available with the 4K Disk Monitor. That is, in addition to the features of ODT which accesses core strictly on an octal basis this package can accept the symbol table output from PAL8 using the /D option and then during your de-bugging it's possible to reference locations by their symbolic name. Unlike the 4K Disk Monitor DDT, however, PSD is core resident and you have to allow space for it. It is based on XDDT (DECUS 8-127) with the addition of OS/8 facilities.

Ray notes that he has had some difficulty with the OS/8 FORTRAN II/SABR package with respect to formatted I/O which he has been trying to implement for the CALCOMP plotter. He is trying to arrange it so that the CALCOMP replaces the high speed punch as device #2 for formatted output. He would like to see more convenient "hooks" into the format conversions for user written routines. He would like to have an arrangement where a user written module is called on a character by character basis for Non-DEC supported device numbers. He has also found a problem in IOH. It seems that during E format output there is a call to the DIV routine, then there is a call to the PRINT routine and then there is a call to the IREM routine to fetch the remainder of the last DIV call. In the nomal DEC I/O routines this proves to be no problem because the I/O routines do not call DIV themselves so its status is preserved, however, in his case his CALCOMP symbol plotting routine is using the DIV routine for scaling purposes. Anyone else trying to write their own special formatted I/O routines could run into the same difficulty. He suggests either storing both exponent digits before outputting either which means a modification to IOH.SB or else to modify IOH.SB to branch to a routine called USRIO instead of GENIO when the device read or write number is greater than 4. If you are interested in contacting Ray he is at MIT Laboratory for Nuclear Science, Cambridge, MA 02139 - Room 24-004.

NOTE FROM H.S. HOPKINS

Mr. Hopkins has made several recent submissions to the DECUS Library. The first is ALPHA which is a versatile sorting program enabling the listing of OS/8 file directories on any of the 4 fields; file name, extension, creation, date and/or starting block number. He would eventually like to be able to integrate this entire facility into DIRECT but at the moment he doesn't have sources to make that possible. At the moment it's a stand alone program. He has also submitted a version of CCL that implements an ALPHA command that works the way a DIRECTORY command would, only it calls his program instead. He has also submitted a version of MARK12 to make it more useful for both DIAL and OS/12. He also has worked up a one-word patch to DIAL to make it work with the LS8F Centronics printer, and possibly the DECwriter IIs on PDP-12's. He submitted an SPR on the problem and DEC's response eventually was that they would not make the fix generally available so he has included the information in his submission to DECUS. He says in anticipation of the delays in getting the programs into the DECUS catalog if anyone needs them he will be happy to be contacted directly. He doesn't have high speed reader and punch so he would prefer to deal either with DECtape or LINCtape for the OS/8 programs and in LINCtape for the PDP-12 programs. He also has a number of things that he has done with DIAL that are yet to be submitted to DECUS which DIAL users may find of interest.

NOTE FROM DAVID MILLER

David suggests a couple of names under our Name-the-SIG project. The 12 Bit SIG might be called BUG BYTES or DE-BUG BYTES. Any more ideas out there or any thoughts on the subject?

NOTE FROM BRIAN CONVERSE

Brian writes to note that in the latest release of OS/8 (OS/12) that he received, 4 of the 8 LINCtape handlers have disappeared from the non-system device LINCtape handler. I have come across that complaint from others also. Does anyone have any information on this question?

NOTE FROM STUART DOLE

Stuart tried the patch to OS/8 BASIC submitted by T. Wes Sikes and discovered that it had problems with long programs on machines larger than 8K. He has sent along a revised patch which fixes the problem and also fixes an "EN" error bug.

NOTE FROM L. E. BYRD

Mr. Byrd notes that with his LA30 printer the previously published patch for DIRECT that changes the default number of columns in a listing to three would not work because three columns were still too wide for his printer. He suggests changing location 123Ø7 from 7125 to 7126 to change the default number of columns to two which he says is quite satisfactory.

PROBLEM USING VT50 TERMINALS

Two people have written with problems that they are having with the VT50 terminal. Both problems are due to the fact that the VT50 seems to send 7 bit ASCII with the 8th bit set to zero rather than the traditional convention on PDP-8's of setting the 8th bit to a one. Some software apparently still doesn't know enough to force the 8th bit on or otherwise deal with the problem. First, Rudi Stange from DEC Sales Support in Munich has noted that DEC's RKØ5 disk formatter program (DHRKØ-A) will not work on a DEC Data System 310 which uses a VT50 as its terminal. This is because it does not handle the 8th bit correctly so it can't accept inputs to the questions that set it up. Therefore, on the system he is working with he cannot format disks. He has sent along a suggestion on how he patched software to make it work. If anyone is interested let me know. Also on the same problem Jim Van Zee notes that the PARAM program in DECsystem-8 has the same problem. He used FUTIL to search through the program for all the occurrences of test constants and he added 200 to them. This should be fair warning to anyone writing software for the 8 that it had better be able to deal with either convention for the 8th bit for the foreseeable future if they want it to be portable from one configuration to another. The most common way to do that is simply to mask each input character with a 177 and then add 200 thus forcing the 8th bit on in the input driver. Then the conventional PDP-8 coding using the 8th bit set will work in the rest of the program.

NOTE FROM ANGUS FERGUSON

Mr. Ferguson sent along a modification to FOTP, some comments on LIBSET and also a version of LIBSET that he has been working on. First, he finds that the /Q option in FOTP is annoying with a LA36 DECwriter due to the fact that the last character of the file name extension is hidden by the printer head before it moves aside after printing the file name and question mark. This slows down the use of that option. The following patch is suggested to add an extra question mark to pad out the position of the head

```
. GET SYS FOTP
. ODT
14233/477Ø 4337
14337/xxxx  Ø; 477Ø; 1371; 4770; 5737
↑C
.SAVE SYS FOTP
```

Notes on LIBSET. The documentation in the OS/8 software support manual pages A7 and 8 for the structure of the FORTRAN II library files is in part incorrect according to Angus. The main error is that each entry point is allocated only one loader control word (LCW) which is always followed by a zero (not two LCW's as is shown on page A8) thus a .RL file with 3 entry points will have 3 entries of the entry point names and load pointers (3x4 word) in the directory and only one LCW no matter how long the file is (less than 37 pages, of course!). Another error concerns the use of the /S option as described in the OS/8 handbook (page 4-69). The option *</S works correctly for the first line (LIB8.RL default output file), however *LIB8.RL </S returns immediately to the command decoder.

The example on page 4-70 of

```
*ASIN,ACOS
*/S
```

also returns immediately to the command decoder after the carriage return on the second line. He says the reason is perfectly obvious if one examines the source. A simple patch of 12616/5244 5225 will fix first bug but a source change is required or a large patch to correct the second bug.

Angus has modified LIBSET considerably, now calling it LIBFII, to do the following:

a. Accept a library file as input on any command decoder line by using the /L option. This provides a method of listing existing library files.

b. All entry points in the new library are listed at the termination of the program.

c. Corrects the above problem in LIBSET.

He plans to submit LIBFII to DECUS. However, if anyone would like a binary copy of it in the meantime they can write to him. The address is: School of Earth Sciences/Dept. of Geology, University of Melbourne, Parkville Victoria, Australia 3052.

He has also forwarded a copy of LIBFII on paper tape to me.

If you like to bomb directories, (or even if you don't like to, but do,) there are several techniques of directory restoration available. First, though, a few comments on how directories get lost.

1. The most straight forward method is to use PIP and do *DEV:</Z inadvertantly. Of course, you realize that to zero a directory requires only three (3) words to be changed in block 1. Thus an inadvertant zeroing of a device by PIP could be undone by a 3 word patch with FUTIL. But this was too simple for PIP. It thinks it has to write all 6 blocks with what amounts to gibberish, thus removing any chance of restoring the original directory. WHY ??

2. Similar to the technique above is the CCL technique of .ZERO DEV:. CCL then invokes the method above for zeroing the device. The original release of CCL (version B) did not check for a colon, so if one did a .ZERO DEV without the colon, CCL obliginqly decided that you wished to zero the default device. So, slam, bam, thank you, there went device DSK:, (which is often SYS:), even if you only wanted to initialize a freshly formatted tape or disk. CCL is smart enough to reject any specification with a filename, (which would be the case if the colon were inadvertantly ommitted,) as a filename with or without a device specified is irrelevant in a ZERO operation. A modification to CCL to provide this check was published in DIGITAL SOFTWARE NEWS of 1974 August. Do it!

3. FOTP or DELETE with *.*/D will leave you with one EMPTY for each directory block used. You get what you asked for, using the method you asked for. The method of deleting all files one by one is not amenable, and need not be, to 'second thought', or failsafe operation.

4. Directory problems can be caused by an i/o problem with the device or media (disk or tape), e.g. checksum error, making a directory block unreadable by the OS/8 device handler. A quick solution that sometimes works in this situation, (especially with LINCtapes with the TU55-TU56 skew problem), is to read the bad block with FUTIL. With most devices the block has been transfered into the memory buffer before a checksum error is detected and the device handler takes the error return. If so, FUTIL will give notice of a read error, but the contents of the block are in FUTIL's buffer. Check it out. Sometimes there are no errors, sometimes one or two to be patched with FUTIL. In any case, a FUTIL WRITE operation after the READ will often fix the checksum problem.

DIRECTORY RESTORATION : If the directory is not up to date, or needs modification, appropriate changes must be made with each technique. Finding lost files requires some detective work - generally with the aid of FUTIL or STECO.

1. If you have a program that saves a copy of the 6 directory blocks in a safe place, it should have a restore mode. Use it, and you're home free.

2. If you have a copy of the directory on paper only, then you have the choice of using FUTIL to generate the whole directory, or zeroing the directory and using PIP with the DEV:name</I=n construction. Both of these techniques are tedious and prone to errors, particularly if one wishes to preserve the file dates.

3.   If there is no copy of the directory preserved, either on paper or in a file, there is little choice but to use STECO for the ASCII files, (see OS/8 Handbook), and FUTIL for the rest.

4.   The FIXDIR protocol has the following salient features :
        1. Most of it is done with ASCII files, so it is versatile and
                easily modified.
        2. It uses only CUSPs. (FUTIL is a CUSP.)

USING FIXDIR :

The FIXDIR protocol uses a DIRECT /E/B output file for directory backup. Of course, this file should be saved on a device other than that for which it is a directory. The DIRECT /E/B output file (Device.DI), is an ASCII file, so it may be readily examined by editors or listers. Any neccessary modifications may be made with an editor. An additional benefit of using ASCII files is that SRCCOM may be used to compare the backup DIRECT file with a DIRECT file from the final restored directory.

STEPS IN USING FXDIR :

1. Locate the backup file and edit it to look like the desired directory for the device.

2. Run TECO, load FXDIR.TE, setup input and output files, execute FXIDIR.
EXAMPLE:

```
.RUN DEV:TECO
*ERDEV:FXDIR.TE$YHXDERDEV1:DEV2.DI$EWDEV1:DEV2.PA$MDEX
```

3. Prepare the target directory area to look like the following :

```
.R FUTIL
OPTION DEVICE DEV

LIST OCTAL 1.0-6

0001.00000:   7777  0000  0006  0000  7777  0000  7777

LI OC 2.0-6

0002.00000:   7777  0000  0006  0000  7777  0000  7777

LI OC 3.0-6

0003.00000:   7777  0000  0006  0006  7777  0000  7777

LI OC 4.0-6

0004.00000:   7777  0000  0006  0000  7777  0000  7777

LI OC 5.0-6

0005.00000:   7777  0000  0006  0000  7777  0000  7777

LI OC 6.0-6

0006.00000:   7777  0000  0000  0000  7777  0000  7771

^C
```

The directory should now look like this :

```
.DIRECTORY DEV:/E/B

 31-DEC-79

<EMPTY>   0000   1
<EMPTY>   0000   1
<EMPTY>   0000   1
<EMPTY>   0000   1
<EMPTY>   0000   1
<EMPTY>   0000   7
  12 FREE BLOCKS
```

4. Assemble the file produced by FXDIR. DON'T FORGET TO USE /F! Restore the directory by saving the file on the device.

```
.R PAL8
*DEV/F/L$
.SAVE DEV:DUMMY
```

If you wish, you may save the file and then load it th[illegible]ugh PIP :

```
.SAVE DV:DEV
.R PIP
*DEV:DUMMY<DEV.SV/I
```

5. Compare the resultant directory with desired results.

```
.DIR TM<DEV:/E/B
.COMPARE DEV.DI,TM.DI

No Differences!
```

The following should be obvious to the most casual observer :

1. The preceding setup will overwrite the bootstrap block, (block 0), and will restore only 5 of the 6 directory blocks. The bootstrap may be restored by using PIP /Y. If you have enough files to use 6 directory blocks, the preceding scheme can obviously be extended to use block 7.

2. If your directory backup file is not /E/B, you must fix it up to look like one or merely make the obvious changes required in FXDIR.TE.

```
! CMPRS.TP 76.02.05 !!
                    TECO MACRO COMPRESSOR :

        The importance of the comment field of a program increases as the
program becomes more complex.  This is perhaps especially true for TECO
programs (macros), which are very compact statements & tend to read as
hieroglipics anyway.  While many languages have assemblers or compilers
which, in effect, strip off the comments, TECO, being an interpreter,
keeps them as part of the program, thereby occupying some of the space,
(which is so often minimal), and slowing execution, (as the comment
fields still must be scanned).

        Comments in TECO are also labels, so, even if it were possible
to strip off the comment fields by syntactical analysis, labels
also would be deleted and the program rendered non-functional.

        CMPRS is a TECO macro that defines all comments that include a
CRLF to be comments, (and therefore expendable), and all which do not
contain a CRLF to be labels (which therefore must be kept).
Thus CMPRS deletes from the macro all comments which include
a CRLF, and deletes the following CRLF if there is one.

        Since a complete syntactical context check is not done, any
literal text fields which contain an exclaimation point must        '
observe even parity within the line of code.  See CMPRS itself
for examples of methods of doing this!!  Of course CMPRS
can compress itself - (with this many comments, it'd better be able to)!

!
Start at the beginning & find the 1st label or comment
(include a dummy exclaimation point for parity.)
The search also provides the means of exit from the loop
!
J < S!$ ^^!$
!
Back off to just before the exclaimation point
If find end of label before find CR, do the whole thing over
!
-1C 0UA < %AA-^^!"E QA+1C ^^!$ OAGAIN$
!
If the next char is CR, then
!
' QAA-13"N >
!
Delete all chars from the begining of the comment to the current location
then delete all chars until find end of comment
!
' QA+2D <0A-^^!"N D ^^!$ >
!
If a CRLF immediately follows the comment, delete the CRLF also
!
' 1D 0A-13"E 1A-^^
                  "E 2D
!
Do the whole operation until no more label/comment fields are found,
then tell'm it's compressed
!
'' !AGAIN! > ZJI
!COMPRESSED!$
!
And that's all there is to it!

 !
```

A postscript on conventions: I like to use the extension .TP (TECO Program) for the commented files, then the standard .TE is used for the compressed versions.

TIM CLARKE
FRELAN ASSOCIATES
BOX 298
MENLO PARK, CALIFORNIA 94025

This is the result of CMPRS.TP being operated on by CMPRS.TP: i.e. the macro itself c̄ no comments

```
! CMPRS.TP 76.02.05 !
J < S!$ ^^!$
-1C 0UA < %AA-^^!"E QA+1C ^^!$ OAGAIN$
' QAA-13"N >
' QA+2D <0A-^^!"N D ^^!$ >
' 1D 0A-13"E 1A-^^
                   "E 2D
'' !AGAIN! > ZJI
!COMPRESSED!$

!COMPRESSED!
```

ITEMS FROM LARS PALMER

__ETOS Users__

As an user of the ETOS Time Sharing System in an environment slightly different from that which it was intended (the educational market), I would be very interested in getting contact with anybody that runs ETOS under similar conditions. We will probably find that we have several things of interest together and that we could share a good deal of experience of the ETOS usage. Anybody interested in such a swap of ideas please send your name and address to me together with a short list of what you can contribute to the ETOS environment. Myself, I have the following material available:

1) TRUE patches for several of the OS/8 V3C programs to run them under ETOS. These patches are TRUE overlays, i.e. they do not reorganize core the way ETOS source patches do and it is therefore possible to use the patching mechanisms supplied by DEC for further patching in the programs. I have patches for most of the cusps, for FORTRAN IV System and for the OS/8 BASIC System except for the Editor which we do not use.

2) I have subroutines available which makes it possible for a FORTRAN IV program to determine whether it runs under ETOS or not.

3) We have added to the TIME program a new option, /A (account) which will give the account and console number of the specified job. This makes it possible to obtain this information from an non-privileged users job. As I understand SYSTAT will not be privileged in next version of ETOS but until then this is a good way around it.

Concerning the IOT codes on PDP-8 I have run up against another one of DEC's internal communication problems. All documentation to PDP-8 says that "FPP 12 has device code 55 (and 56) and AD converter device code 53". This is so in all new configurations and all DEC software including MAINDECs are assembled with these codes.

We received a new PDP-8A Lab machine and decided to test it with FORTRAN IV before installing the AD8A which was placed in the bus as per delivery. Everything crashed. This was repeated every time we ran FORTRAN but all MAINDECs (except the AD converter which was not yet tested) ran perfectly. On a thorough digging into the documentations to the machine, we found a sentence saying "The AD8A is delivered with the device code 55 installed" (the AD8A like the KL8E has switch selectable device codes so fixing the problem was trivial once the problem was diagnosed).

When doing sampling from ADC to disk there exists a need for fast routines. Many attempts have been made to achieve a better result than the standard REALTM WRITE(n) combination. I have just completed one routine WRITBL/READBL which I have sent to Bob Hassinger. The following information might be interesting.

I have tested several routines for thruput speed. The configuration used is PDP-8E + RK8E + FPP12. The actual test programs used are available from me or Bob Hassinger. Note that these times probably represent minimum speeds for the various configurations. If you allocate larger buffers you can probably get better results but I wanted to test the most difficult combinations.

The programs are:

1) THRUPUT comes from the TSAR package. It replaces REALTM and is much faster as the data is not floated but passed on to the disk with 255 AD values/ block. Floating is then achieved when the data is accessed (in non time critical environment).

2) WRITEB by R. Phelps also achieves a more compact saving of data by re-fixing the data before writing to the output file.

3) WRITBL I just wrote. Recognizing that much of the time is spent in passing a value to the output buffer, I wrote a routine that instead passes the output buffer address to FRTS. The restrictions on its use are not very severe and it works fine. I will put it on the DECUS tape sometime but at present you can obtain it from me or Bob.

The results:

	Max sampling speed possible		
Routines	FPP	No FPP	
REALTM+WRITE(n)	800	%	
REALTM+WRITEB	2300	%	WRITEB is in FPP code.
REALTM+WRITBL	3000	200	
THRUPUT+WRITE(n)	2400	2000	
THRUPUT+WRITEBL	4000	3400	4000 is the limit for clock service in FRTS

% = about or LT 100

FROM DAN SMITH

I've been working with FORTRAN IV on a PDP-12 without FPP, and am passing along some notes about things which do not seem clear in DEC's documentation, in hopes of speeding other peoples' learning process. Assembly-language (RALF) programming within the FORTRAN system is almost hopeless without full listings of 1) FRTS, and 2) the non-arithmetic library routines. I think DEC should either include these in the software support manual, or make them available as a reasonably-priced package separate from the rest of the FORTRAN source.

I would appreciate any comments or corrections. In particular, I would like to know if there are any techniques suggested here which would not work on a system equipped with FPP.

1) Debugging

On a PDP-12, a very useful technique for debugging RALF code is to set an E-stop at location 00040. These are the low 12 bits of the simulated FPP program counter. With the stop set, (which should be done after the program has begun initiation--it is helpful to begin the program with a READ(4,format)DUMMY), repeatedly pressing CONTINUE will "single-cycle" through the RALF code. The memory buffer may be read as the address of the instruction being executed; that is, one should wait until the address has advanced just beyond the current instruction before examining memory or FLAC for its effect. One can freely use examine and deposit, and still have the program continue normal execution when the E-stop is removed; however, for reasons that I don't understand, fill step and step exam should be avoided; I suspect that they destroy some internal registers or status information that are needed to proceed correctly after an E-stop.

2) Normalization

It is sometimes desirable to write machine-language programs that generate floating quantities; for example, a data-acquisition program that generates a file structure that will later be read by a FORTRAN IV unformatted read. For this and other reasons, it is useful to know the floating-point format, and, in particular, how normalization is handled.

FORTRAN IV arithmetic routines assume that the operands are normalized, and may produce incorrect (not just inaccurate) results if this is not the case. (For example, the floating add assumes that a quantity must be zero if the high half of the mantissa is zero.)

Unnormalized quantities can be generated by ordinary FORTRAN code in several ways: by the use of Hollerith variables, by unformatted reads, or by calling RALF subroutines. Fortunately for those of us who like to manipulate Hollerith, ordinary data transfers (e.g. I=J) do not perform normalization. In fact, it is rather tricky to force a normalization in FORTRAN code; Q=Q+0 will not do it. One should never attempt arithmetic on unnormalized quantities; this includes the formatted output routines, which do ugly things if the quantities presented to them are unnormalized. (Note that nonzero integers within the FORTRAN system are, of course, normalized).

The condition for normalization is that the mantissa M satisfies 1/2 .LE. M .LT. 1. This is almost, but not quite, equivalent to saying that the two highest bits must differ. There are two exceptions, both involving negative numbers: 60000000 is a legal normalized mantissa, and 40000000 is not. It is important to avoid creation of the quantity 40000000. It is not sufficient to dump an unnormalized quantity into the FLAC and do an FNORM, because it turns out that FNORM does not properly handle 40000000.

3) Use of FLAC from 8-mode sections

Although it is not mentioned in the RALF manual as a way of communicating between RALF and 8-mode sections, when running under FRTS, the FLAC is located at 00044, 00045, and 00046. Thus 8-mode sections can access or alter the FLAC. I believe this would be true when running with an FPP as well, because the APT should be saved and restored from the same locations; that is, I believe this technique isn't mentioned in the RALF manual because it is FRTS-dependent, not because it won't work with an FPP. Can anybody out there confirm or deny?

4) Watch out for autoindexing!

I've been burned a couple of times by forgetting that a SECT8 section can load anywhere--specifically, it can load into page 0. Thus, any SECT8 section should either be checked carefully to make sure that no indirect references are made in locations 10-17 (and that includes JMP%'s, folks), or it should be forcibly prevented from loading into page 0 by making it a FIELD1 section and including something like COMMZ #PAGE0, ORG 200 just before the FIELD1 declaration.

5) Operand locations in STARTD mode

The formulae on p. 5-8 of the OS-8 handbook don't seem to fully specify exactly which words constitute the operand, particularly in STARTD mode. If Y is the operand address, as computed from the formulae, then in STARTF mode, the operand in all cases consists of words Y, Y+1, and Y+2. In STARTD mode, however, it consists of words Y and Y+1 when the instruction is double-word direct reference or single-word indirect reference, but consists of words Y+1 and Y+2 when the instruction is a single-word direct reference. This scheme makes sense, because it permits the address part of the indirect address pointers to be loaded or stored by single-word direct references, but it is not immediately obvious from DEC's description.
Notice also that base-page offsets are always multiplied by 3, regardless of mode.

6) RALF addressing pitfalls

Problems arise if RALF elects the single-word direct addressing format when the programmer does not expect it. Two likely examples:

a) The programmer is in STARTD mode and expects the operand to be at Y and Y+1. Because the single-word format is used, it will actually be at Y+1 and Y+2.

b) The difference between the address, A, and the base page origin, B, is not an exact multiple of 3. In this case, RALF happily computes an offset of (A - B)/3 with no warning or error message. The effective address will be at A - (A - B mod 3), which is not what the programmer expects.

For an example of these problems, see the original coding of the FORTRAN IV clock routine, in which the TIME entry point references FLDA OVRCNT instead of FLDA# OVRCNT.

This leads to the interesting question of just how RALF decides which format to use. It is more complicated than the manual implies. A working hypothesis follows: let FINS stand for any of the data reference instructions FADD, FDIV, FMUL, FSTA, FADDM, FLDA, FMULM, and FSUB; let A be a symbolic location on the base page.

a) Adding a # suffix (FINS#) forces the two-word format (as documented on p. 5-22).
b) Adding a ' suffix (FINS') forces the one-word format (is this documented?)
c) If a symbol A is <u>forward</u> <u>referenced</u> by an <u>unsuffixed</u> data reference instruction, then all references FINS A will use the two-word format.
d) If A is not forward referenced, or is forward referenced only by instructions of form FINS# and FINS' (or by 8-mode instructions), references of form FINS A will use the one-word format, and will be correct only if (A - B) is an exact multiple of 3.

In general, it is wise to use the # suffix unless the address is known with certainty to lie at an exact multiple of 3 on the base page. It seems particularly important to use it when using double-precision mode to stuff addresses, instructions, JA's, etc. into the middle of executable code.

A rationale for point (c) above is as follows: during pass 1, forward references must be the long form, because it is not yet known whether A is on the base page and the assembler must make a commitment so that the location counter does not become undefined. It would seem that backward references could still be the one-word form; however, during pass 2 it is difficult to distinguish forward from backward references without complicated symbol-table entries (since all symbols are defined at the start of pass 2); so the simplest workable solution is to flag the forward reference in the table and always use the long form.

It seems to me that DEC should give high priority in their next RALF revision to having RALF either a) use the long form, or b) give an error message, or c) both, when a base-page reference is not divisible by 3. Any of these options is easily implemented with about a dozen extra words of code. Unfortunately, RALF's field 0 is pretty tight right now, and finding those extra words seems unlikely unless some code can be moved to field 1, which is feasible but will take some work.

The following kludgey patch will implement option a) above for users with 12K or more. Warning: it hasn't been texted extensively yet!

PATCH FOR RALF V60A

```
.GET DTA1 RALF

.ODT
change
  6262/0140 3240                               /make patch level "Z" (so won't be
                                               /confused with genuine DEC patches!)
* 1120/4772               jms i (over3
  01121 /1155 6222        cif 20               /replaces tad [200
* 01122 /1046             fpps3, tad opcode /louse up AC if entry from formt2
* 01123 /4560 †           jms i [outwrd        /when entry is from formt2, we'll
* 03041 /0000                                  /jms to 23041 instead of to outwrd
* included for clarity, no change needed
  23000/0067              extmp                /over3 leaves remainder*2 here
  23001 /1061             formt1               /entry point for long form
  23002 /0070             extmp2               /over3 leaves quotient here
  23003 /0200             200                  /constant
  23004 /1122             fpps3
                                               /ignore return address from jms
  23042/7200              cla                  /ac has garbage at this point
  23043 /1600             tad i (extmp         /pick up remainder
  23044 /7650             sna cla              /if it's bad,
  23045 /5250             jmp .+3
  23046 /6203             cif cdf 0            /then return to formt1
  23047 /5601             jmp i (formt1        /with AC clear.
  23050 /1602             tad i (extmp2 /pick up quotient, fortunately still around
  23051 /1203             tad (200             /do instruction clobbered by cif 20
  23052 /6203             cif cdf 0            /and return to fpps3, with AC and hopefully
  23053 /5604             jmp i (fpps3         /everything else just like they would have
  ↑C                                           /been without patch
.SAVE DTA1 RALF  0-7600,12000-13777,23000      /save more than we got!
                (PROGRAM, SYMBOLS,  PATCH)
Change as indicated
```

7) Driving user routines via CLOCK

Although the manual says cryptically that a "common" use of the FORTRAN IV CLOCK subroutine will be in conjunction with REALTM, it does not explain how it can be used for anything else.

The CLOCK subroutine contains an entry point, #CLINT, which seems to be intended for users. #CLINT is a two-word (STARTD mode) block, which initially contains zero. At any given time, #CLINT should contain either zero, or the ADDR of an 8-mode subroutine. The subroutine must reside in field 1, is entered with AC clear and IF=DF=1 via a JMS, and should return the same way (i.e. it should follow the same rules as a subroutine for use with ONQI or ONQB). The subroutine will be called once per clock tick. If no clock subroutine is desired, set #CLINT to zero and then CLOCK will maintain the right time and listen to the schmitts if desired, but will do nothing else.

8) A bug in CLOCK

Which brings up an interesting point. Routine IDOCLK in the CLOCK routine is incorrect, in that if any schmitts are enabled, the #CLINT routine will be called on interrupts from the schmitts as well as from the clock overflow.

Since REALTM works by dropping an an address into #CLINT (#CLINT can only be used to service one routine at a time; needed, and presently under development, is an ONQB-like routine for use with the clock), this means that presently, if any schmitts are anabled, REALTM will sample and buffer the a-to-d's on schmitt events as well as clock ticks, which is probably a bad thing.

The remedy is obvious; just after JMP LOP2 following DOTRIG in the clock routine should be inserted CLA, TAD CSTAT, SMA CLA, JMP% IDOCLK (to skip the call to #CLINT if the clock didn't overflow). This change hasn't yet been tested.

9) More conflicting device codes

So Hans W. Goebel think's he's been had because the FPP-12 maintenance IOT's conflict with his AF04? We've got the same problem, only different; our AA50 D-to-A converter has device code 655x, same as the FPP itself! First, the good news: we don't have an FPP and don't plan to get one. Next, the bad news: FRTS talks to your FPP even if you don't have one, once per interrupt. Next, the good news: a nop at 00407 in FRTS shuts it up.

--Dan Smith
Eye Research Institute
20 Staniford St.
Boston, Mass. 02114
617 742-3140, X 260

Shift/8: A Program to Convert FPP to BCD Numbers

Minicomputer systems usually do process control functions under assembly language programming, while the requirements imposed by the user to do analytic computions normally forces the user to leave the realm of assembly language programming and use higher level languages as Focal, Basic or Fortran. With the help of the Floating Point Package (FPP) and its associated mathematical subroutines, it is possible to do analytic computation in assembly language with comparative ease. The nature of the FPP number, however, does not lend itself well to output. One FPP number uses 3 core locations. These core locations contain the binary equivalents of the exponent and mantissa.[1] When outputting to peripherals, data formats are usually specified by some type of 7 to 8 bit serial or parallel BCD ASCII code. Therefore, all data must be transformed to a properly coded form if it is to be transmitted from mini to a peripheral.[2]

This subroutine provides a method of converting the 27 bit FPP[3] numbers to an equivalent 3 digit ASCII BCD number. All FPP numbers must be normalized to be 999 (base 10) before the call is made to the subroutine. Two features of the subroutine are that 3 word FPP number can be preserved or overwritten by the subroutine is completely relocatable.

The Shift/8 operation is carried out on 3 word FPP data arrays auto-indexed by octal core location 0011 and the coresponding 3 digit BCD numbers are stored, one digit per word, in an array auto-indexed by core location 12. By loading the same data location into core location 11 and 12, the BCD numbers generated by Shift/8 will overwrite the respective FPP numbers. This could be utilized as a core saving feature of Shift/8. The number of data points on which Shift/8 is performed is specified in the location labeled REFC.[4] Shift/8 is self initializing such that one REFC is fixed for a single array, N arrays may be changed by calling Shift/8 N times. Care must be taken in the user program to insure that core locations 11 and 12 have been set with the new array locations.

[1]See Intro to Programming, Chapter 8, p8-1, 8-41

[2]A case in point is the output software handler need for the PDM-70 as indicated in the article, "Interfacing the FDP-8/L to the Laboratory Using A Programmable Data Mover", Decuscope, Volume 14, Number 3

[3]DEC-08-NFPEA-A-PB (1972)

[4]This counter should actually reside in page Ø and be loaded with the number of variables before the routine is called.

CLEARED 28 Aug 75, INFO OFC, ECOM

for [illegible]
JEFFERY A. SLUSHER, INFO Ofcr, NVL

```
.PALD
*OUT-S:SHIFT
*
*IN-S:SHIFT
*
*OPT-I

                              *400
0400  0000  SHIFT,   0
0401  1333           TAD REFC
0402  3332           DCA CC
0403  1411  NUMBR,   TAD I 11        /GET FNUMB
0404  3330           DCA TEMP1       /STORE EXP &HOM
0405  1330           TAD TEMP1       /GET FNUMB
0406  0303           AND MASK1       /GET EXP
0407  7012           RTR
0410  7010           RAR
0411  1326           TAD K12         /NO OF RAR TO SHIFT
0412  3327           DCA STORE
0413  1330           TAD TEMP1       /GET FNUMB
0414  0304           AND MASK2       /GET HOM
0415  7012           RTR             /JUST WITH
0416  7012           RTR             /HIGHEST BIT IN LINK
0417  3330           DCA TEMP1       /STORE HOM
0420  1411           TAD I 11        /GET FNUMB+1
0421  0305           AND MASK3       /GET MOM
0422  7012           RTR             /JUST MOM TO HOM FORMAT
0423  7010           RAR             /PUT DEAD BIT IN LINK
0424  1330           TAD TEMP1       /ADD HOM
0425  7010  ROT,     RAR
0426  2327           ISZ STORE       /INCR ROT POINTER
0427  5225           JMP ROT         /NORMALIZE
0430  0311           AND MASK7       /BINARY<999(10)
0431  3312           DCA INPUT
0432  1315           TAD CNTRL
0433  3240           DCA PNTR        /INIT TABLE
0434  7100           CLL
0435  1314           TAD CNT         /SET BIT 7
0436  3313           DCA NUMB        /STORE ROTS
0437  1312           TAD INPUT
0440  1316  PNTR,    TAD TABLE       /OR INCRMNT
0441  7430           SZL
0442  3312           DCA INPUT       /INPUT+TABLE
0443  7200           CLA
0444  1313           TAD NUMB
0445  7004           RAL
0446  2241           ISZ PNTR        /INCRMNT TABLE
0447  7420           SNL             /FINISHED?
0450  5236           JMP PNTR-2
0451  7106           CLL RTL
0452  7006           RTL
0453  1312           TAD INPUT       /SHIFT LEFT AND ADD
0454  3330           DCA TEMP1       /3 BCD NOS
0455  1330           TAD TEMP1       /GET 3 BCD NOS
0456  0306           AND MASK4       /GET HOBCD
0457  7112           CLL RTR         /POSITION
0460  7012           RTR
0461  7012           RTR
0462  7012           RTR
0463  1331           TAD ASCII       /8 BIT ASCII
0464  3412           DCA I 12        /STORE HOBCD
```

```
0465  1330            TAD TEMP1          /GET 3 BCD'S
0466  0307            AND MASK5          /GET MOBCD
0467  7012            RTR
0470  7012            RTR
0471  1331            TAD ASCII
0472  3412            DCA I 12           /STORE MOBCD
0473  1330            TAD TEMP1          /GET 3 BCD'S
0474  0310            AND MASK6          /GET LOBCD
0475  1331            TAD ASCII
0476  3412            DCA I 12           /STORE LOBCD
0477  2311            ISZ 11             /SKIP ENUMB+2
0500  2332            ISZ CC
0501  5203            JMP NUMBR          /DO NEXT NO
0502  5600            JMP I SHIFT
0503  1770  MASK1,    1770
0504  0007  MASK2,    0007
0505  7770  MASK3,    7770
0506  7400  MASK4,    7400
0507  0360  MASK5,    0360
0510  0017  MASK6,    0017
0511  1777  MASK7,    1777
0512  0000  INPUT,    0
0513  0000  NUMB,     0
0514  0023  CNT,      20
0515  1316  CNTRL,    TAD TABLE
0516  6340  TABLE,    -1440              /-800
0517  7160            -620               /-400
0520  7470            -310               /-200
0521  7634            -144               /-100
0522  7660            -120               /-80
0523  7730            -50                /-40
0524  7754            -24                /-20
0525  7766            -12                /-10
0526  7764  K12,      -14                /-12
0527  0000  STORE,    0
0530  0000  TEMP1,    0
0531  0260  ASCII,    0260
0532  0000  CC,       0
0533  7767  REFC,     7767             /SEE FOOTNOTE #4

ASCII   0531
CC      0532
CNT     0514
CNTRL   0515
INPUT   0512
K12     0526
MASK1   0503
MASK2   0504            PNTR    0440
MASK3   0505            REFC    0533
MASK4   0506            ROT     0425
MASK5   0507            SHIFT   0400
MASK6   0510            STORE   0527
MASK7   0511            TABLE   0516
NUMB    0513            TEMP1   0530
NUMBR   0403
```

FORLIB 'SYNC' EXTENSION FOR DK8-E CLOCK

Ian M. Templeton, National Research Council
of Canada, Ottawa, Canada K1A OR6

The standard CALL SYNC (I,N) returns N=1 if Schmitt trigger I has fired since the last call (I=1,2 or 3). A minor modification allows I=4 to return N=1 if the clock has overflowed since the last call, and thus provides for simple clock synchronisation. One more flag location is provided and precleared on a call to CLOCK, and a 4-word patch is inserted in place of some redundant KW12-A instructions. The changes may be made with EPIC, but the precise locations to be changed must be determined by the user. The only occurrence of CLAB (6133) provides a useful pointer. In the latest version of FORLIB, CLAB occurs at location 337 of block N, and the modifications are made at N(341-2), N(361-75) and N+1(21 & 35). The required changes are listed below. N.B. An earlier version of FORLIB has addresses in SYNC differing by 1. If N(341) contains 1230 instead of 1231 all the commands marked * should be reduced by 1.

```
.R EPIC

*FORLIB.RL</IS
R
S,6133,7777
Ø1Ø7 Ø337
6133 /
O,337
6133 /                  CLAB          /USEFUL SEARCH POINTER
72ØØ /                  CLA
1231 /3227 *            DCA STFLG+3 /CLEARS 4TH FLAG LOCN
Ø364 /7ØØØ              NOP
O,361
1366 /1365              TAD FCNWD   /1,2,3 OR 4 REQUESTED
711Ø /Ø364              AND P7      /MASK TO BE SURE
1365 /1375              TAD KSTFLG+1/FLAG POINTER
743Ø /32Ø4 *            DCA SETCLK  /TEMP STORE
7Ø41 /16Ø4 *            TAD% SETCLK /GET FLAG
Ø2Ø1 /3365              DCA FCNWD   /RETURN Ø OR 1
1375 /36Ø4 *            DCA% SETCLK /CLEAR FLAG
32Ø4 /62Ø3              CIF CDF
16Ø4 /5712 *            JMP% DOSYNC /RETURN
3365 /751Ø  PATCH,      SPA          /BIT Ø = 1 IF CLOCK OFLO
36Ø4 /1324 *            TAD PATCH   /SET BIT 8 IF CLOCK OFLO
62Ø3 /Ø363              AND P17     /MASK AC 8-11
5712 /5353 *            JMP ON
R,11Ø
O,2Ø
1223 /                  TAD ISVBIT
Ø364 /5324 *            JMP PATCH
744Ø /      ON,         SZA
O,34
ØØ17 /      P17,        17
Ø377 /ØØØ7  P7,         7
E

*↑C
```

FIXES OF BASIC V3

THESE PATCHES TO THE BASIC OVERLAYS CORECT A BUG NOTED IN THE DIGITAL SOFTWARE NEWS FOR THE PDP-8, SEQUENCE 21, JAN 76. THEY ALSO ADD A SIGNIFICANT ENHANCEMENT. T WES SYKES SUBMITTED THIS ENHANCEMENT TO THE OS/8 SIG NEWSLETTER NO 15, P17, BUT IT FAILS ON SYSTEMS LARGER THAN 8K.

THE BUG FIXED IS THE "EN" ERROR. THIS PATCH PREVENTS THE SYSTEM FROM CRASHING WHEN THIS OCCURES BY RE-SWAPPING THE 176ØØ PAGE. THE ENHANCEMENT IS TO ALLOW FILE LOOK-UPS WITHOUT ERRORS. IF THE LOOK-UP FAILS, THE END-OF-FILE BIT IS SET SO THAT A "IF END #N GOTO XXXX" CAN TEST THE SUCCESS OF THE OPERATION AND ALLOW THE PROGRAM TO CONTINUE.

```
.GET SYS:BASIC.FF

.ODT

1431Ø/4516 5774 (JMP I XPATCH .. WAS ERROR CALL)
14374/XXXX 3544 (XPATCH, PATCH .. ADDRESS OF HOLE FOR PATCH)
13544/XXXX [illegible] (CIF 1Ø .. START BY CALLING THE USR)
13545/XXXX 4477 (JMS I USR)
13546/XXXX 11   (=USROUT)
13547/XXXX 4556 (JMS I P1SWAP .. RESTORE PAGE 176ØØ)
1355Ø/XXXX 7344 (SET AC=-2)
13551/XXXX 1755 (TAD I XFNOM ... GET THE USR CODE THAT CAUSED ERR)
13552/XXXX 765Ø (SNA CLA .. WAS IT LOOK-UP OR ENTER?)
13553/XXXX 5543 (LOOK-UP, SO SET EOF BIT, NO ERROR)
13554/XXXX 4516 (ENTER, A HARD ERROR, BUT NO CRASH)
13555/XXXX 43Ø5 (XFNOM, FNOM ... ADDRESS OF USR CODE ARG)
↑C

.SA SYS:BASIC.FF

.GET SYS:BASIC.SF

.ODT

12635/3467 4223 (-NEW ERR CALL-1)
↑C

.SA SYS:BASIC.SF
```

FROM: Stuart Dole
1386 HSE
Univ. of Calif.
School of Medicine
San Francisco, Calif. 94122

digital

#18 24

SPR #

Field #

Page of

SOFTWARE PERFORMANCE REPORT

Name: U. VAN ZEE (THAT'S 'J' AS IN 'JIM')	Software Specialist
Company: DEPT. OF CHEMISTRY BG-10	Office / Cost Center
Address: UNIVERSITY OF WASHINGTON / SEATTLE, WASHINGTON Zip 98195	Report type: [✓ Logic/coding error] [✓ Documentation error] [✓ Suggestion] [✓ Inquiry]
Phone: (203) 543-2576 — Date sent: 26 AUGUST 1976	
Computer: PDP-12, ETC. — System device: RK05 — Memory: 8K	Other hardware: LINCTAPES
System program & version: PAL8-V10 — Monitor & version: OS/8-V3C	Document: OS/8 HANDBOOK — Page: 3-12

Attachments: ☐ terminal printout ☐ object tape ☐ source tape ☐ listing [✓ example]

NEW VERSION OF PAL8 HAS VASTLY REDUCED SYMBOL TABLE - HENCE CAN NOT ASSEMBLE MANY PROGRAMS WHICH CAN BE ASSEMBLED WITH V9. HANDBOOK STATES 992 SYMBOLS/4K (FIELDS 1 AND ABOVE) WITH 897 ON A 8K SYSTEM. I OBTAINED 894 WITH V9, BUT ONLY 636(!!!) WITH V10. THE TEST PROGRAM CONSISTED OF A FILE CONTAINING TAGS: SYMØØØ, ... SYMxxx, ETC. THE NUMBERS QUOTED ABOVE WERE THE NUMBERS PRINTED AFTER THE 'SE' ERROR MESSAGE, HENCE I MIGHT BE OFF BY 1 DEPENDING UPON WHE~~T~~HER THE ERROR REFERS TO THE PREVIOUS SYMBOL OR THE ONE THAT OVERFLOWS.

HAVE DISCOVERED THAT V10 WILL AUTOMATICALLY USE ADDITIONAL CORE (SO DON'T NEED TO SPECIFY /K) - THAT'S NICE BUT HOW ABOUT ALL US USERS WITH 8K SYSTEMS?? WILL YOU CONTINUE TO SUPPORT BOTH VERSIONS?

SUGGESTION FOR A MINOR IMPROVEMENT TO BOTH VERSIONS: ADD A FILE COUNT ERROR MESSAGE SO THAT AN INADVERTANT '$' IN A FILE BEFORE THE LAST ONE WON'T LEAD TO INCORRECT ASSEMBLY WITH NO INDICATION OF AN ERROR (ASSUMING THAT LATER FILES ARE OVERLAYS, ETC. WHICH CONTAIN NO FORWARD REFERENCED SYMBOLS - A COMMON SITUATION). MY IDEA IS JUST TO COUNT THE FILES FROM THE CD INPUT, DECREMENT AT EACH EOF AND ERROR OUT IF $=Ø

FOR SOFTWARE SPECIALIST USE ONLY		FOR SOFTWARE COMMUNICATIONS USE ONLY	
DATE REC'D	TO S.C.	DATE REC'D	TO MAINTAINER
DATE ANS'D	TO CUSTOMER	DATE ANS'D	TO SPECIALIST

#18 25

28776

digital **SOFTWARE PERFORMANCE REPORT**

FIELD #: | SPR #:

FOR DEC USE ONLY

Page ____ of ____

SYSTEM PROGRAM AND VERSION (OR DOCUMENT) FORTRAN IV 3.05	MONITOR AND VERSION OS/8 V3C	DATE 76-07-13

NAME: Lars Palmer

FIRM: AB HÄSSLE

ADDRESS: Fack
431 20 MÖLNDAL 1
SWEDEN

ZIP

SUBMITTED BY: PHONE:

LIST ATTACHMENTS

DEC OFFICE

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
8E	512	RK8E	32	

Compiler – Assembler does not detect a memory overflow in a Dimension statement. See enclosure.

```
 CORE
32K CORE!

 EX TEMP.FT-T/F/T‡  OS/8  FORTRAN IV  3.05              JUL 13  1976

0002                DIMENSION A(4000),B(4000),C(4000)
0003                DATA A.B.C/12000*0/
0004                DO 10 I=1,10
0005       10       WRITE(0.100)I
0006       100      FORMAT(I6)
0007                END

NO ERRORS
21 SYMBOLS, NO ABS REFS

 #       C 00000   #BASE     00023   #EXIT  X 00000   #GOBAK    00057
 #G0001    06400   #G0002    06404   #LBL     06370   #LIT      00075
 #MAIN   S 57030   #NE     X 00000   #RENDO X 00000   #RET      00015
 #RSVO   X 00000   #ST       06370   #TMP     00064   #WRITO  X 00000
 #XR       00002   #10       06406   #100     06431   A         00130
 B         27470   C         57030   I        00061

INPUT ERROR
```

SOFTWARE COMMUNICATIONS USE ONLY

DO NOT PUBLISH ☐	DATE RECEIVED	BACK FROM MAINTAINER	LOGGED ON
	TO MAINTAINER	DATE CLOSED	LOGGED OFF

SOFTWARE COMMUNICATIONS