



DECUS 12 BIT SPECIAL INTEREST GROUP
NEWSLETTER

January

Number 26

1978

Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator
12 Bit SIG
c/o DECUS
129 Parker Street, PK-3/E55
Maynard, MA 01754

..or.. Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

DECUS/Europe contributions are solicited through:

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

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

NEWSLETTER SUBMISSIONS

The Newsletter is currently published bi-monthly in the odd months. The deadline for each issue is the last Friday of the preceding even numbered month. Submissions are accepted at all times and are normally used in the next issue to go to press regardless of date of receipt. The deadline for ready-to-use material for the next Newsletter is 24-February-1978. Material requiring editing/re-typing should be in earlier. Ready-to-use material should use an area 6 1/2 inches (16.5 cm) wide by no more than 9 inches (23 cm) long on each page. It should be single spaced on white bond paper whenever possible and must be reasonably clean, legible and sufficiently dark for good photographic reproduction.

THE DIGITAL SOFTWARE NEWS SNAFU

As many of you knew, a large number of PDP-8 and PDP-12 users stopped getting the Digital Software News for the PDP-8 in the last few months for some reason. No one know why and everyone both inside and outside of DEC was very confused about what was happening. The last few months I have been writing nasty letters and beating on everyone in sight about the problem. We finally have an answer !!

Angela Cossette, who is in charge of Software Communications at DEC (the people who manage the production of DSN but not its content), finally realized there was a real problem and went to work on it. She found

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

that there had been a super EDP foul-up last April that no one had noticed. It seems that her data base of names and addresses for the people who receive DEC's various DSNs and Software Dispatches has a special gimmick in it to override the normal automatic removal of a name at the end of a year in the case of PDP-8 DSN. The intention is that people remain on that list forever. The special gimmick interacted with the merging of two different parts of the data base in such a way that the few thousand names that are also on DECUS' PDP-8 mailing list had the delete flag set for April. Due to personnel changes, etc., etc., no one noticed that all those names disappeared from the mailing list. Gary Cole and Angela have assured me that there was no intention to remove anyone from the list and the policy remains as always to continue PDP-8 DSN service indefinitely for everyone.

I understand that the the problem has been found and fixed. Everyone should start to receive the DSN again monthly and I was told that Software Communications plans to send out a summary of everything that was missed to everyone affected. Score one for the SIG !

SESSION REPORT - 12 BIT SHORT NOTES

The following is a transcription of notes taken by Bill Lennon during the Fall Symposium in our the 12-Bit Short Notes session. Thanks Bill - we need volunteers to take notes in all our working sessions so the rest of the SIG can benefit from the Symposia sessions that are not recorded in the printed proceedings.

The informal session was designed to allow brief time slots for anyone who wanted to talk about what he had been doing or discuss a problem he was interested in.

Larry Alber from the Food and Drug Administration gave a brief talk on how he has connected his PDP-12 and PDP-8e and the RTS-8 tasks he has written to use the link. He also talked about an RTS-8 analog to digital converter task he did for the system.

Tom McIntyre talked about the West Virginia University program package. Because Tom is leaving WVU to work for DEC and so will be unable to continue maintaining the programs, he plans to submit them to the DECUS Program Library soon. Tom wanted to try to set up a user supported maintenance system for the package. He suggested that it could be the prototype for more extensive SIG support of user written, general purpose software.

We agreed that initially I (RH) will act as a clearing house for coordinating the effort. Bill Lennon offered to get someone at Northwestern University to maintain the MEDIA package. Jim Coryell has been working with SORT for some time so he and I will start the support for it. We need volunteers to help expand this project. Proven, well written and maintained software of general interest has a chance of being distributed with OS/8 in the future (as in the case of FUTIL in the Extensions kit this time).

Bill Lenon reported on the Magtape work being done at Northwestern University and asked for input on the general "ANSI labeled tape" question.

Tom McIntyre initiated a discussion on a suggested project to convert the FORTRAN II library to MACREL/LINKER. The more difficult question of how to convert the entire FORTRAN II system from SABR to MACREL was discussed. This would be a very worthwhile project if it could be done.

I initiated a discussion on the facilities and resources needed at the Symposia for media conversion, distribution and exchange of software. Bill Lennon volunteered some student help during the Chicago meeting. A committee was formed to work on the project. A letter from the committee asking for input is included in this Newsletter.

John Alderman announced that his DSP-8 Diagnostic Support Package for the PDP-8 has been converted into MACREL (mostly as 'RSECT's), described the package and offered to exchange copies during the meeting with those interested.

There was discussion of Bob Phelps programs and FORTRAN IV routines. TU10 and TM8e controller fixes were also discussed.

FUTIL

It is now official. Jim Crapuchettes' FUTIL is being distributed in the OS/78 and OS/8 Extensions V3D kits. User documentation is included in the OS/78 manual and the OS/8 Handbook Update for V3D.

DEC describes this very useful CUSP thus:

"This CUSP allows examination and modification of the contents of mass storage devices. Uses include patching programs that contain overlays, examination and repair of OS/8 directories, bad block checking and correction, decimal/octal conversion of double-precision numbers, outputting the Core Control Block of .SV files and the Header of .LD files, and creation of special directories."

I believe that the distributed version also supports MACREL/LINKER type .SV files with overlays. In fact FUTIL is the only practical way to patch such files. The standard ODT system will not work if the file has overlays.

Jim and the SIG have been working on getting a user written CUSP accepted into a DEC distribution for many months. At the Fall Symposium the indications were that DEC felt this project had been a success and they were interested in continuing it with other software. I will be persuing the subject to see what we can accomplish. With DEC and the users working together this way we can make the 12 Bit software better than either the users or DEC could alone.

NEWSLETTERS ON MICROFICHE

DECUS has suddenly found out about the wonders of microfiche. They seem to be seen as the answer to many of the printing and mailing costs problems. For example the primary distribution of the new PDP-11 Library Catalog was on fiche with an option to send back the fiche in return for a printed copy. At the Fall Symposium there were even suggestions to put the original Newsletter mailings on fiche. The advantages would be lower cost, the ability to include more pages, maybe faster reproduction, lower postage cost and so at least a chance mailing might be done first class to speed up distribution. Fiche viewers are getting fairly cheap, they can usually be found in libraries and they are usually available in organizations of any size. Hard copy can be made from fiche at many libraries and other suitably equipped facilities.

I pointed out that a full switch over to fiche might present considerable problems for some 12 Bit SIG members so it should not be done without a careful study of the impact on the membership. However, I suggested the following possible course of action as an interim measure. We could offer Newsletters in both fiche and hard copy forms. The first reproduction priority would be the fiche with the printed copies (which probably take longer to produce) being done afterwards. If economics permit, we would mail the fiche first class (in the US at least) and continue to mail the paper copies as they are now. This would offer improved service as an inducement to use the fiche and at the same time those taking the hard copy version would continue to get just about the same service they now get. We would have to see what the numbers and costs were for those who want both forms or multiple copies before we would set a policy on that subject. It may be that we would have to charge a fee for those cases. How do you feel about this subject? Do you like the idea of trying to improve service and reduce costs through the use of microfiche? What impact would it have on you?

NEWSLETTER BACK ISSUES AND INDEX

The question of back issues of the Special Interest Group Newsletters has been a hot topic the last year or so at DECUS. There has been an attempt to charge prohibitive fees for back issues this year. That policy was set in an ill advised manner without adequate consideration of the impact or implications. The US Publications Committee first requested then demanded a change to the policy. No formal action was taken but I understand that an informal study showed that the charges were not worth the trouble. At the moment the picture is still cloudy but at the Fall Symposium it seemed that as long as the demand remains low, the US Executive Board will not require its members to pay for back issues. The policy remains fluid and the position of the other chapters is unknown at this time.

An alternative to hardcopy of back issues (which is fairly expensive to produce and mail) is microfiche. It should be possible to put all 25 back issues of the 12 Bit SIG Newsletter on a very few fiche that would be much easier and cheaper to reproduce, handle and distribute. I am exploring the possibility of doing this for our Newsletter as a trial case.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 26 - January 1978

Dan Smith has been hard at work indexing the back issues of the Newsletter and has all but three of them done (the index is already about 8 pages long). I hope that we will be able to do the microfiche experiment and include Dan's index. If not I will try to publish it in the Newsletter. Your inputs on these subjects are solicited.

FOCAL STANDARDS, DISASM AND FORTRAN IV OPTIMIZER

Earl T. Ellis recently wrote about the work he is doing with the newly revitalized FOCAL SIG. In their session at the Fall Symposium it was decided that the first thing to do was to develop some sort of standards for FOCAL so different versions will evolve in as compatible a form as possible. I plan to send Earl the copy of PFOCAL he requested. Anyone else who has information about a version he is interested in should do the same.

I tried the FORTRAN IV optimizer on my PDP-8/I and got a 400% speed increase for the example he gives. I also tried it on the Whetstone benchmark and found almost no difference at all. If your program does a lot of certain things, the optimizer could be useful. It is nicely designed so it fits into the F IV system and works automatically.

My latest tape from Jim van Zee has a copy of the newly typed source files for DISASM and SPLIT. It has always been hard to get a source of these programs, so the maintenance that was needed never got done and I had trouble using them as a result. They are probably the best tools available for taking apart a program when you only have a binary or save file. They let you iteratively disassemble sections of code, assign labels and so on, and produce a resulting source file that can be assembled.

December 12, 1977

Dear Sir,

I am writing as a results of conversations at the FALL DECUS. I have sent my listing of DISASM to Jim Van Zee, along with what I had typed-up. I am getting that back now. I am also working on Standards for FOCAL for the SIG. For the standards, I would like a copy of the original OMSI FOCAL, which everyone seems to have copied for OS/8 support. I could buy it from DECUS, but the paperwork would take months. Fred Strange and I have agreed the standard should be for a 16K OS/8 configuration, like the DECSTATION 78, with 8K and Standalone as Subsets, and the LAB Functions, Plotting, and EAE routines as Supersets. How much to include from FOCAL for the 11,10, and 15 has not been determined.

I have U/W-FOCAL, FOCAL-69, and EFOCAL (a var of GW-FOCAL (a var of P-FOCAL)). I am currently getting 32K-LINK-FOCAL (DECUS Poster Board), FOCAL/F from Georgia Tech, and GW-FOCAL from George Washington University. I would welcome input from the SIG.

We have two PDP-8/Es. both have 32K, TD8-E, HSR, three RK05 drives, XY8 Plotter, and RS232 Lineprinters. One system will soon have a 9-Track and several PCM-12 microcomputers to control instruments remotely. The other system has a HSP, VC8-E, and AD8-ES with 2 AM8-EA's. As yet, there is no link between the two systems. Both 8's have ETOS, OS/8 through FORTRAN-IV (currently V3C). We also have INBASIC, LAB-8/MS, U/W-FOCAL, EFOCAL, LAB-BASIC, SCROLL, RUNOFF, DIRECT.5G, and FUTIL7. the PCM-12's also run FOCAL in 4K, 8K and 16K in stand-alone mode. Most of our CPU time is spent using U/W-FOCAL.

The enclosed tapes is not blank. It contains an F-IV Optimizer by Eric Swanson of the University of Washington. Try is with the following program. I was amased.

```
                SUM=0
                DO 20 I=1,10000
                X=ABS(I-5000)
                SUM=SUM+X**2
20              CONTINUE
                WRITE(4,100) SUM
100             FORMAT(E16.6)
                END
```

I hope to submit it to DECUS for him in the near future. He has a 32K PDP12 with EAE, RK05, and FPP. He said submission was OK if anyone is interested. What is your opinion.

You can put any of this in the SIG. Especially the call for input. My address is:

```
                USCG R & D Center
                  Avery Point
                  Groton, CT 06340
                (203)445-8501 ext 296
```

Thank you very much,

Earl T. Ellis Jr.

BAND-AIDS FOR THE DATE

Reverend Chase sent a couple of patches to help people keep going after the first of 1978 using old versions of OS/8. They are only intended to tide you over till you get OS/8 V3D up.

To change CCL to print correct dates in range 1978-1985:

```
.R EPIC
#SYS:CCL.SV</1$
R,12
O,366
0003/ you type 6 and a carriage return
```

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

0,373
6760/ you type 6770 and a carriage return

W
E
* type control-c to exit

Notes:

- 1) The 3 to 6 change corrects day of week printouts.
- 2) The 6760 to 6770 changes the base year from 1970 to 1978.
- 3) The two changed locations are literals. In many CCL versions they will be where they are shown here. In other, user-modified dialects, you may find them 1 or 2 locations away from the locations shown (366, 373). But they have to be in this general area -- somewhere !
- 4) With this patch you enter the date with 8 subtracted from the year. For example:

.DATE 1/1/70 to enter 1/1/78

This fake is required only when setting the date; date print-outs (from .DATE command) will be correct (1978, etc.).

- 5) If you start the day with ".R 1978" (see below) or its equivalent ODT patch (i.e. .OR. the 0200 bit into the contents of location 07777 to set the extended year bits) then programs that know about the new date scheme in OS/8 V3D (such as DIRECT V5, COBOL-8, U/W FOCAL, etc.) will print correct dates.

/ '1978' PROGRAM TO FIX UP EXTENDED DATE BITS PAL8-V9H 12/31/77 PAGE 1

/ '1978' PROGRAM TO FIX UP EXTENDED DATE BITS
/ OF VIRTUAL MACHINE ('ETOS'), OR OF
/ STAND ALONE MACHINE (OS-8).

0000	FIELD 0	
4000	*4000	
04000	6201	CDF 0
04001	7300	CLA CLL
04002	1777	TAD I (7777
04003	7040	CMA
04004	0376	AND (7577 /.NOT. 0200
04005	7040	CMA
04006	3777	DCA I (7777
04007	7600	7600
04010	5607	JMP I .-1

/.SAVE SYS 1978;4000=6403

04176 7577

04177 7777

\$\$ PAS 12/77 \$\$

TECO SIG

I recently received the first issue of the TECO SIG Newsletter ("The Moby Munger" would you believe !). In spite of the name, it is full of interesting information about all versions of TECO running on 12, 16 and 32 bit DEC machines. The editor is Stan Rabinowitz who has been a leader at DEC in the effort to bring all DEC's versions of TECO as close together as possible. Except for the PDP-10 where there is no active maintainer to help move that version towards the standard, Stan has had a large measure of success. Between the PDP-8/12 and PDP-11 versions there is now a high degree of compatibility. TECO is thus one of the most "portable" tools available on more than one computer family. If I get users to learn TECO (or a subset), they can go between our 8 and our 11 with hardly a thought and almost no additional training (the similarity between OS/8 and RT-11 concepts helps with this also). Others have been doing the same sort of thing between an 8 and a PDP-10 for several years. They say sometimes they have to look twice to be sure which terminal they are using!

At the Fall Symposium the newest PDP-11 version of TECO with extra features for support of the VT-52 CRT terminal was running. By loading a TECO macro that uses the VT-52 support features, you get a very nice scope oriented editor with special function keys that is like a superset of TECO on a PDP-12 (i.e. with refresh display) or PDP-11 TECO with a VT-11 refresh display system. If the talk at the Symposium about putting the same VT-52 support functions in OS/8 TECO should come to fruition, the same scope editing macro can be used and exactly the same scope editor will be available on both 8's and 11's. That should be a very popular item. It might even tempt me to swap my old teletype for a VT-52 on our PDP-8/I.

If you are interested in TECO and its applications, its development, extension and standardization, and/or TECO programs (macros) you should be a member of the TECO SIG. The last 12 Bit SIG Newsletter had a form to use to join the TECO SIG. If you do not have a copy you can ask DECUS for one.

DIRECT V5

Jim van Zee sent a note giving the results to date for his inquiry in the Newsletter regarding distribution of the V5 (user enhanced) version of DIRECT. In spite of the fact that the questionnaire was not easy to remove and fill out so letters had to be written, Jim had received 46 replies by November 22 with 10 of them from OEM customers. About one third of all the replies (about fifteen) were from outside the U.S.: Argentina, South Africa, Canada, Germany, England, Holland, Australia and New Zealand.

Forty-four out of forty-six wanted the enhancements in DIRECT V5 made available in some way. Forty out of forty-six thought putting binary paper tapes in the Newsletter was a good idea and several of the

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 26 - January 1978

remainder who did not indicate support for that idea said they only had DECTape and they would be willing to trade DECTapes with anyone. Jim notes that "curiously there were no responses from floppy disk people". Those replying offered to prepare about 670 copies of the paper tape for Newsletter distribution but that is not enough to proceed in that direction (Newsletter circulation is between 2000 and 3000 and there are some questions about the cost and mechanics of this sort of insertion). Jim is going to send paper tapes to those who responded, however.

Jim notes that among the responses "were a half-dozen interesting comments among the letters as the authors expressed (generally) dissatisfaction over their relationship with DEC". Jim suggests that "it might be useful for someone in Maynard to get in touch with a few of these people to see if there is any substance to their feelings".

During the Fall Symposium the DEC representatives seemed to be pleased with the results of the experimental inclusion of FUTIL in the new release of the OS/8 Extensions kit. As a result there seemed to be considerable interest in exploring the possibility of doing the same for one or more additional user written programs in a future release. I am following up with DEC on this subject and will report in the Newsletter as progress is made. One of the high priority items to consider for inclusion is the enhanced version of DIRECT. I think that this is the best hope for getting widespread distribution for DIRECT V5. Until something comes of these efforts, however, the DECUS Program Library plus direct user exchange can do a reasonable job of distributing it to those who really want it most.

DECUS PROGRAM LIBRARY NEWS

Recent Submissions:

CHECMO II: Chess Playing Program - DECUS 8-822

This is John E. Comeau's chess playing game for the PDP-8 family. Versions have circulated among users for some time but it was only recently submitted to DECUS. As far as I know this is the best chess game readily available at this time for the 12-bit family of computers. Good players can certainly beat it however. I understand that some work has been done on a bigger, more sophisticated chess program for the 12-bit family but nothing has been said publicly so far. Incidentally, the chess game often seen running on PDP-11's at DECUS Symposia is said to have been derived and extended from this program.

Price codes: Write-up - A01, Binary Paper Tape - F06, DECTape - H12, a copy on Floppy Disk will soon be available under code K09.

DECSYK - DECUS 8-872

PDP-8/E programs to allow a Sykes 7100/7200 Floppy Disk system to read and write DEC Format Diskettes. Uses the MQ register and a 6400 word buffer. Needs an alternate file-structured device such as DECTape to move the files to.

"The data on a DEC diskette is written in 3/4 filled, interleaved sectors. It appears to be impossible to unscramble these via an OS/8

handler within the timing restrictions of the unbuffered Sykes system, so the present approach has been used. The 'READ' program reads sectors sequentially, two tracks at a time, and deposits the data in the correct (interleaved) places in a 6400-word buffer (in field 3, but this can be changed). This buffer is then copied, via a standard OS/8 handler, to an alternate file - structured device. As written, the program uses the (already resident) system handler with unit set to 2, and writes the copy on a blank diskette, but any appropriate variation of this may be used. The 'WRITE' program does the opposite, first filling the buffer from the source device, and then copying that to the DEC diskette in such a way as to reproduce the correct interleaving while writing sequentially."

"Because the DEC diskette contains 494 blocks = 76 tracks, (with track 0 unused) the final 'READ' or 'WRITE' operation will leave the Sykes head automatically stepped to the (non-existent) track 77 position. The resulting error message is ignored, and the head is stepped back by one track to avoid subsequent hangup. The final halt occurs with DF=IF=0, so (after removing the DEC diskette!) a system diskette can be restarted at once at 07605."

"Both 'READ' (DTOS) and 'WRITE' (STOD) programs make extensive use of the MQ register. It is not certain that alternate (slower) operations will necessarily meet the Sykes timing criteria."

Price codes: Write-up and Listing - D01, Floppy Disk - K09.

CARD.BA - CARDIAC Simulator - BASIC8-103

Needs 16k of memory, OS/8 BASIC V3.21, Extensions to BASIC DECUS 8-860, and a Card Reader is highly recommended. Program is designed to use a VT-50 or VT-52 as console.

"CARD is a comprehensive simulator for CARDIAC, a simplified machine language developed by Bell Telephone Laboratories for teaching elementary programming concepts. CARD, written in OS/8 BASIC, interprets and executes CARDIAC programs, while also providing editing, tracing and listing capabilities. Input comes from cards or keyboard and output can be directed to the console or line printer. Fully interactive and diagnostic it is easy to use and very useful in educational environment. Complete instructions and four sample programs are provided. Assumes a VT-50 or VT-52 terminal and requires 16k words of memory."

Price codes: Write-up and Listing - D01, Floppy Disk - K09.

MTFOTP - MAG-TAPE FILE-ORIENTED-TRANSFER-PROGRAM

While at the Fall Symposium I recieved an updated rundown on Bob Phelps' programs. One that I don't think has had much mention before is MTFOTP.

"This program allows reference to files on a mag-tape using a directory similar to OS/8 directories. The program works to and from mag-tape in exactly the same way as FOTP works to and from any other directory device. In addition, MTFOTP has three additional features:

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 26 - January 1978

/P -- This switch causes the mag-tape to be positioned before the specified file. The file can then be read by any program as if it came from a non-file structured device.

/S -- Use sub-file directory on mag-tape.

On output, the output file name (or the first input file name if no output name is given) is the only entry made in the directory. All input files are copied, and their names are entered into a "sub-file directory".

On input from mag-tape, if /S is specified, the first input file name is the name of the sub-file directory and is not transferred per se. The remaining input files are then transferred from this sub-file, or if no additional files are specified, all files in the sub-file directory are transferred.

/Z -- Zero the mag-tape directory.

=NN - This option transfers NN files beginning with the first file in the input list if input is from mag-tape. Subsequent files are those in sequential order immediately following the specified file.

The command "DIR MTAO:" will cause listing of the mag-tape's directory. The command "DIR MTAO:<sfd name>/S" will cause listing of a sub-file directory from mag-tape."

Bob's address is Behavior Laboratory, Dept. of Radiation and Biophysics, University of Rochester Medical Center, Rochester, NY 14642.

LETTER FROM JIM VAN ZEE

22 December 1977

Dear Bob,

Since the last Newsletter indicated that approximately 50% of the people requesting new DECUS catalogs were using FOCAL, I thought that a brief description of some of the new features I have added to my version (U/W-FOCAL) in the last few months might be of general interest. This is not meant to be another round in the great 'FOCAL vs. the WORLD' debate, but merely a discussion of some interesting ideas which I think tie up a number of 'loose ends' in the structure of the language.

The first innovation is the introduction of 'relative line numbers', following a suggestion from Doug Wrege at the June DECUS meeting, and a little push from Dan Smith. Line numbers such as '.51' are interpreted to mean 'line XX.51' in the current group. This convention saves a great deal of program space, particularly in IF or ON commands, and makes code 'group independent' so that you can put a little subroutine in any group you like without searching through

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

it to change all the line numbers. It also speeds up execution somewhat since there is less text to scan and fewer digits involved in an ASCII-to-binary conversion.

Since such numbers obviously belong to 'Group 0', this group is now available for program use, but with one peculiarity: commands in Group 0 can only be executed when the program is started from the beginning, since a command such as 'GOTO 0.75' will simply transfer to line XX.75 in the current group. Only by restarting the program with a GO or DO command can you get back into Group 0 after slipping out of it. This makes this group convenient for 'once only' code or for comments about the program as a whole. Incidentally, direct commands are considered to belong to 'Group 0' so you can edit or call any line in this group directly - an obvious requirement!

At the same time that I let Group 0 in, I also removed the restriction on lines such as '1.00'. These lines suffer from the difficulty that a command such as 'ERASE 1.00' will do the usual thing and erase the entire group, but nevertheless, it seemed to me that it would be nice to have the first line of each group available, at least for use as a COMMENT line. This is also convenient when using computed line numbers to index into an entire group.

The second major innovation is the addition of NEGATIVE line numbers, such as '-1.5'. Such numbers are interpreted by commands like DO, WRITE and ERASE to mean 'start here and finish the group'. Commands such as GOTO or BREAK make no distinction between negative and positive line numbers. The utility of this feature is immediately obvious: it permits multiple entry points for subroutines, such as 'DO 5' or 'DO -5.2', as well as permitting things like listing just the last half of group 9 with a 'WRITE -9.5' command. Incidentally, the line specified does not actually have to exist; it only serves as the starting point for a line number search.

This feature is essential, in fact, for erasing Group 0, since an 'ERASE 0' command would delete the entire program whereas 'ERASE -.01' will only remove lines in the range 00.01 to 00.99. Another convenient use of this feature is for combining several small subroutines into a single group, using a RETURN command to limit the range of the call. I refer to commands such as 'DO -8.4' as 'sub-group' calls. They may occur in LIBRARY GOSUB, ON and JUMP commands as well as in FOCAL Statement Function calls. Relative line numbers may also be negative, so a call such as 'DO -.8' may be used to execute a short subroutine at the end of the current group.

But the biggest reward comes when constructing multi-line loops. Since FOCAL's loops extend only to the end of the line following the FOR command, loops which involve more than a few simple commands have always had to be written elsewhere in the program and called with a 'DO'. This leads to the problem of where to put the subroutine? If you use the very next group for it, you succeed in keeping the program fairly well organized, but at the cost of continually having to jump over the body of the loop when you are finished. On the other hand, if you put the subroutine out of the way near the end of the program,

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

you do a rather grave injustice to the idea of 'structured programming'.

The solution offered by the 'negative line number' feature is to allow one to write 'vertical' loops in FOCAL just as you would in any other high-level language. The form of such loops is:

```
1.1 FOR I=1,N; DO -.2
1.2 first line of code
1.3 second line of code
1.4 ...
1.6 last line of code
1.7 NEXT
```

This will loop through the 'sub-group' consisting of the lines immediately following the FOR command, being terminated by the NEXT in the last line. Upon completion of the loop the program will then resume with the statement following the NEXT command (i.e. the 'next command!') unless this command specifies a different (non-zero) line number. Of course the NEXT could go at the end of the 'last line' rather than on a separate line as shown above - the choice is more one of personal taste than anything else. Clearly this makes translation of programs written in FORTRAN or BASIC a very simple matter which could, in fact, be done almost automatically.

The third major innovation is to add a line number option to the RETURN command. This allows 'multiple returns' as well as 'multiple entry points' for subroutine calls. This was suggested to me several years ago by John Cowan, but it did not seem practical at the time. However the rewrite to include the features just mentioned left sufficient space to include the return option, a true case of serendipity! The command 'RETURN 5.1' will abort the usual return sequence and transfer instead to line 5.1. This means that you can selectively turn 'DO's into 'GOTO's, a feature particularly welcome for use with the ON command. Of course relative line numbers are useful here too, so 'R .7' will abort the subroutine call and continue at line XX.70 in the current group. This also works with LIBRARY GOSUB calls so you can remain in a sub-program if you want to, but the line number option is always ignored by FOCAL Statement Functions since it is not legal to interrupt an arithmetic evaluation in this manner! As is usual for such commands, a value of '0' denotes 'no branch', so you can even use a computed line number to determine whether you will return to the calling command or continue elsewhere.

Well, these are the three main innovations which could be easily transported to other versions of FOCAL. They seem to me to complete a number of ideas I have tried out over the years, such as the NEXT and JUMP commands, for instance. JUMP, which was at one time a 'computed GOTO' has now been turned into something much more useful: a 'computed DO', since this always seemed to be how I needed to use such a command. But without the option to abort the return, I would get caught in situations where the use of a DO call was a disaster. Similarly, the loop structure outlined above became possible as soon as the NEXT command was perfected, but it was ungainly because each line in the body of the loop had to be separately executed as 'DO 1.2,1.3,'

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

etc. Clearly relative line numbers just by themselves would help, but the 'start and keep going' feature really brings it all together.

Another improvement which has some utility is a 'USER' command which lets you restart the program at a specified point upon detecting the error of your choice. This permits typing out a message such as: 'NEGATIVE ARGUMENT IN THE SQUARE ROOT FUNCTION IN LINE 12.9' rather than just aborting with the message '?21.57'. The recovery technique is not very general since the stack is reset, but it is better than nothing. A better approach would be to initiate a 'DO' call, but that gets pretty complicated since the state of the interpreter is not well defined under error conditions.

I also figured out a neat way to include the '@' character in text strings without having it print in places where you don't want it, such as in the header line or during a symbol table dump. And I added video terminal support so that 'rubouts' generate the usual 'backspace, space, backspace' sequence necessary for updating the display on 'glass teletypes'. The current version runs happily in the background under RTS8 as well as when called by BATCH, so FOCAL jobs can now be mixed with other OS/8 activities. I also included the somewhat controversial LIBRARY ZERO command for zeroing the directory on any device using the length specified by the command. It has the virtue that it only changes 4 locations in the first block (unlike good old PIP which really wipes things out) so you can recover via FUTIL (or FRA) if you need to. It also preserves the system area if there is one. My principle requirement for such a command is so I can easily initialize both short and long LINCtapes, but others had requested such a feature for use with data analysis programs. New manuals and summary cards are currently being printed and will be sent (as promised!) to all 'official' users. I would like to take this opportunity to apologize to all those who have written me without receiving a reply. This is a 'one-man' operation here and my correspondence is at least 6 months in arrears.

We received our DSD floppies last week so I can now convert from almost anything to anything else, with the exception of DECassettes. Mark Champion in our electronics shop also just finished installing a version of the DECTape bit-swapper on our PDP12 which Tom McIntyre developed a few years ago. This is a super nice addition to the TC12-F option which permits true OS/8 operations on DECTapes with all the convenience of LINCtape. It is just like adding another device to the machine since you can run FOTP, make changes with FUTIL, write parameter blocks, etc. Mark simplified the design somewhat, found a slot in the main backplane, and put the whole thing together in about two days for a cost of \$50 or so. I then reworked Tom's handler to allow for two entry points (DTA0: and DTA1: naturally - that ought to end the debate about calling LINCtapes 'DECTapes'!) and removed a serious timing problem I discovered in the original implementation. If anyone out there in PDP12-land is interested, just drop a note to Mark, %Dept. of Chemistry; University of Washington; Seattle, WA 98195, and he will send you a copy of his drawings and my version of the 'DT12' handler.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

Incidentally, the version of DIRECT described 2 Newsletters ago (V5G) does not misbehave (so far as is known) when the parameter block gets clobbered by the directory. This bug was never reported to me - the 'fix' was quite accidental!

Sincerely,

Jim van Zee

FORTRAN IV ACCURACY QUESTION

Anand Kumar writes "We have been using FORTRAN IV on a PDP-8/e with FPP but due to some hardware problems the FPP breaks down very often. The software emulator for the FPP in FORTRAN IV does not optimize calculations accurately. This we have seen by using the same data with and without the FPP. The 4th digit is not accurate without the FPP."

"I wonder whether others have noted this problem. I would like to hear about it. If this problem is not known then I would like this to be signaled to other users who do not use an FPP but still use FORTRAN IV."

The address is Lab for Psychofysiology, Universiteit van Amsterdam, Eerste Constantijn Huygensstraat 20, Amsterdam.

INDUSTRIAL BASIC NOTES

Michael E. Mazzone has sent another set of inputs on OS/8 Industrial Basic. To conserve space I will condense them here and suggest that you write to me or Michael for copies of the actual patches he refers to. He is President of Process Control Systems, Inc., 18130 S. Thornapple Lane, New Berlin, Wisconsin 53151.

In response to an inquiry in the newsletter about the prospects for swapping the function overlays in OS/8 BASIC and I/O, Michael responds: "... DEC Industrial Basic in the core only mode reserves areas in field 1 and field 2 for overlays, and swaps them from this high core to field 0 when required. It is lightning fast compared to DECTape or floppies. Although there would be some work involved in converting OS/8 BASIC, at least it has been done to a similar product."

"OS/8 Industrial Basic offers some help for the second problem, too. The system is interrupt driven with ring buffered terminal I/O. This product is described in DEC Software Product Description 5.10.2. The DEC manual for INBSIC is "Industrial Basic Language Reference Manual" (DEC-S1-OSIBA-B-D)."

I will include Michael's letter on the features of INBSIC as compared to OS/8 BASIC elsewhere in this Newsletter.

On the matter of the "date problem": "This patch to INBSFN.OV, the string function overlay for the Industrial Basic Runtime System, will allow the DAT\$(X) function to return date words corresponding to 1978 and 1979:

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 26 - January 1978

```
GET SYS INBSFN.OV
.ODT
22263/4267      7000    /Prevent using ASCON routine
22300/2700      6770    /67=7, 70=8, i.e. 78 is the base year
^C
.SAVE SYS INBSFN.OV
```

When 1980 dates are required, make this change:

```
.GET SYS INBSFN.OV
.ODT
22300/6770      7056    /70=8, 56+ the bits from the OS/8 date
                /word will give 1980-1985
^C
.SAVE SYS INBSFN.OV
```

Unfortunately, the first patch will only work for 1978 and 1979 dates, and the second will work only for 1980-1985 dates. Due to the lack of space in the string function overlay, only 8 locations spread over 3 pages, there is not much hope of making a more intelligent fix."

Michael sent a patch to the Industrial Basic Runtime System that "... turns off the UDC (Universal Digital Controller) or ICS (Industrial Control Subsystem) digital outputs whenever an Industrial Basic program encounters a fatal error or ends normally. For some reason, the DEC version just left every output in its current state when a program ended, which in some cases could do some real damage to personnel and machinery."

Also included were a set of power fail/auto restart routines. "The Industrial Basic Runtime System (INBRTS) as a standard detects a power failure with the SPL iot (6102). On power up, the present program is not restarted, and a chain is done to a file 'POWERUP.BA'. This is fine in many cases, but in just as many this is an extremely inconvenient way to handle a power outage, especially when the outage is of such a momentary nature that it doesn't even unlatch motor starter coils."

"So, these routines were developed to provided an instant restart on power up following a power failure. The routines are identified by what type of clock is on the computer, since different initializing code is required by each clock."

"Any potential user of the instant restart feature is cautioned that:

- 1) Not every process lends itself to an instant restart after an indeterminately long power failure. Be very sure that no danger to life, property, or product results from such a restart.
- 2) The system clock will, of course, be incorrect for interruptions of more than one second.
- 3) A power interruption during a file operation will probably cause problems with the data being transferred.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 26 - January 1978

- 4) These routines as written do not tell anyone that the power has failed and a restart was accomplished. This could cause problems in an unattended process.

All these routines use space in field 0 which is not currently used by INBRIS."

INDUSTRIAL 14 PROGRAMMABLE CONTROLLER

In the material submitted by Michael E. Mazzoni for the last two Newsletters he has discussed DEC's I-14 line of products. He felt that it was being phased out even though it is still viable. I was therefore interested to note at the Fall Symposium that the Traditional Products Group has taken over the I-14 line and was promoting it with a new full color flyer. Considering inflation and TPL's low volume, higher cost manufacturing, the price increases that Michael mentioned are not too surprising. I wonder if TPL will be able to see to it that software support for the I-14 is maintained on the 8 and 11?

DISKLESS DISK MONITOR

T. J. Miles sent me a copy of a set of overlays he has done that convert DEC's 4K DISC/DEctape Monitor to run on a core only system. He substitutes extended memory for the disc. Since the 4K Monitor was designed to work with a 32K DF32 disc, this could give a fairly good way to use a PDP-8 or PDP-12 that has a large core memory but does not have a suitable system device for OS/8 or one of the other mass storage oriented monitors. The overlays were developed from core dumps because he does not have access to sources or listings. Incidentally, he is doing the work on his own PDP-8/m which apparently does not have a disc. I will try to get the programs into the DECUS library. He would also like to get in touch with a user of TSS-8 (i.e. EduSystem 50). His address is 3970 Saanich Road, Victoria, B.C., CANADA, V8X-1Y6.

OAK RIDGE NATIONAL LABORATORY

OPERATED BY
UNION CARBIDE CORPORATION
NUCLEAR DIVISION



POST OFFICE BOX Y
OAK RIDGE, TENNESSEE 37830

December 15, 1977

Mr. Robert Hassinger, Coordinator
12-BIT SIG
Liberty Mutual Research Center
71 Frankland Road
Hopkin, MA 01748

Dear Mr. Hassinger:

At the Fall-Winter DECUS meeting a 12-BIT SIG committee was formed to help with user program exchange at the next DECUS meeting. User's wishing to use this service can send programs to one of the following committee members.

Best regards,



Russ Overbey
Fusion Energy Division

DRO:rap

Send LINC TAPES - Larry Alber
FDA Room 1222
433 W. Van Burren
Chicago, IL 60607
(312) 353-5863

DEC TAPES - Earl T. Ellis, Jr.
MAG TAPES USC&G R & D Center
DEC FLOPPIES Avery Pt.
Gruton, CT 06349

DEC TAPES - Russell Overbey
DEC FLOPPIES P. O. Box Y Bldg. 9201-2
AED FLOPPIES Oak Ridge National Laboratory
Oak Ridge, TN 37830
(615) 483-8611 Ext. 3-5176

12/20/77

ANNOUNCING

Formation of a Micro-8 Working Group
From Jonathan Lockwood (305) 724-7542 MS 54-40

In the last newsletter (No. 25, which I haven't received yet - three cheers for DECUS) I introduced myself. Since then, I have been to the fall Symposium and there discovered a growing interest in the DECstation 78 and microprocessor based PDP-8 systems. To provide a focal point for such systems, I am forming a Micro-8 working group under the 12-bit SIG. The goals of this working group will be to share ideas, problems and solutions to problems about systems using the 6100 microprocessor. The first DEC system using this microprocessor is the DECstation 78 with its blossoming array of applications. In addition, there are several users that have built dedicated systems for data acquisition and/or process control. These users have severe power/size constraints and thus can not use a PDP-8M or other such large mini computers.

Fall Symposium

I was surprised to learn that there are only 2,200 people on the 12-bit SIG mailing list when there are over 40,000 PDP-8's in the field. If only half of that number were active members, just think what great and wonderful things we could get done.

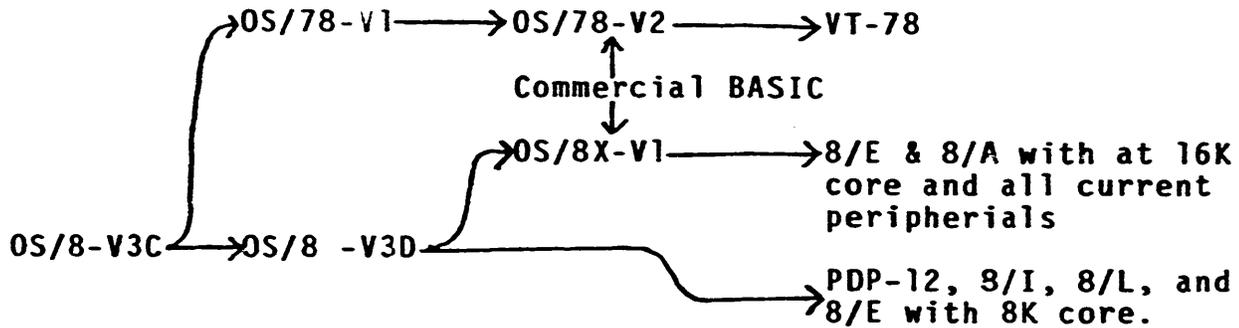
During the PDP-8 product panel Gary Cole, the PDP-8 product manager, said that DEC is working on a commercial BASIC package for the DECstation 78. This package will have random access of files, i.e. it will be able to access individual records (max. record length = 1 block) and it will allow use of key words. It will have more accuracy for string arithmetic. It will use 8 bit ASCII throughout. It will have better print controls.

Next Mr. Cole talked about the new operating system, but he made no firm commitments. He said that the next version of OS/8, which he called OS/8X, would probably require a minimum hardware configuration of 16K of core. Several reasons were given:

- 1) The existing version is almost out of room;
- 2) It would include commercial BASIC which needs more core than the present BASIC;
- 3) Handlers are getting bigger and thus need to use high core;
- 4) The maximum core is now up 128K and thus needs some support routines.

It will also have a new manual (finally)!!

For the older machines, Mr. Cole said that OS/8 - V3D will still be around for a while.



On the second day of the symposium Gary Cole held a session on DECstation 78 applications. He said that 90% of the current applications are for the commercial world. Several groups within DEC are offering packaged systems. One of these is Word Processing:

- WT-78 - Word Terminal 78, no floppy disk.
- WS-78 - Word Station 78, with floppy disk.
- DX/RSTS-WT78 - multi-user system using a PDP-11 as central computer.
- WS200 - multi-user system PDP-8/A as central. Uses a VT52 instead of a WT-78.

Another package is for business products:

- COS/310 - DIBOL operating system for PDP-8/A and RX-11.
- COS/308 - DIBOL system for a VT-78
- WD-78 - Word Data system 78, combination of word processing and DIBOL system (a nice bundled package).

Finally there is the OEM version:

- 78/10 - Bare terminal, no floppy disk, not for sale yet.
- 78/40-DA - Terminal, dual floppy disk, desk, and OS/78

There was a desire to bundle OS/78 with Word Processing to save the User's some money, however DEC is not sure of this for now. Mr Cole showed the following production schedule for the DECstation 78:

	Sept'77	25 units
	Oct	100 unite
	Nov	125 units
	Dec	200 units
Ultimate goal	-	700 units/ month.

Gary Cole said that the reason why the DECstation 78 is not for sale as just a terminal is that you can't do anything with it, i.e. you need either floppy disk or the EPI (Electronic Program Injection) module to boot strap from. Mr Cole then proceeded to "dream" about an EPI that could allow a terminal only version of the VT-78. The EPI would make the VT-78 operate as a "universal smart terminal". This terminal would have standard timesharing characteristics so that it could be used with existing systems. The software could emulate several keyboard operations to "fake out" the host machine and thus provide extended functions such as a whole file transfer with only one keystroke by the operator. It could provide a down line loading capability. It would provide intelligent remote I/O functions such as a plotter. Someone suggested that the existing cassette protocols could be adapted for such use.

Finally Mr. Cole talked about communication options for the DECstation 78. Currently you can use the two RS-232 ports with either the DX protocol for word processing or the DECRET-8 protocol for transfers to the PDP-8's. He said that DEC is considering a synchronous interface adapters for the DECstation that would allow the 2780, By-Sync, and .X25 protocols to be used.

During the open discussion period, several people showed interest for an IEEE 488 interface bus as an adapter for the DECstation 78 and as an interface to the OMNIBUS on a PDP-8. This would make a great paper for the spring symposium in April.

Spring 78 Symposium

I will be co-ordinating one or more sessions regarding Micro-8 applications. If you have any applications that you want to talk about, let me know. If I hear about your applications from other people, I'll come looking for; we need your support! My initial idea is a session early in the week that would begin as a product panel on DECstation applications, similar to the one described above, and then lead into user applications. Another session may be needed if several stand alone system applications are found, such as the one about "Twenty Thousand Leagues Under the Sea with a Micro-8 computer". The drop dead date for the call for papers is Feb. 1st, so let know soon.

New Micro-8 Hardware

Several new products have been announced that will help user's to build all CMOS Micro-8 systems. The most important of these is a set of Tri-state CMOS bus drivers that have a propagation delay of only 50ns with 300pf load. Several pinouts are available:

- HD6431 hex latched input;
- HD6432 hex bidirectional;
- HD6433 ad bus separator;
- HD6495 hex buffer.

Also there is a HD6440 a one of eight decoder with latched inputs. There are several varieties of 4K CMOS RAM available.

FROM DAN SMITH
Eye Research Institute
20 Staniford Street
Boston, Mass. 02114
617 742-3140

#26 - PAGE 22

ANOTHER MICRO HEARD FROM

In #24, I briefly described four microcomputers with the PDP-8 instruction set (Intersil's "Intercept" series; Automated Data Systems' "MICRO-8;" Pacific Cyber/Metrix' "PC/M-12A;" TLF, Inc.'s "MINI-12"). DEC's own VT-78 is, of course, also "a microcomputer based on the Intersil 6100." I also mentioned Cybertek, which does not make complete systems, and noted that their low-power cards made a battery-powered PDP-8 feasible.

Zonge Engineering and Research Organization, 5634 East Pima, Tucson, Arizona 85712, 602 885-3478 has, according to a letter just received, been marketing a battery powered PDP-8 for over a year. The fancy version is said to draw less than 1.5 watts, and the letter implies that their machine has seen successful use under actual field conditions.

Jim van Zee says I should have mentioned Harris as a second source for the 6100 chip.

NEWSLETTER BACK ISSUES

Recently, a friend sent me copies of newsletters #2-16. There is a definite need to make back newsletters more easily available to new SIG members. There is a great deal of relevant and valuable material in back issues. Without distribution of back issues, the newsletter can document the reinvention of the wheel but not prevent it.

For example, like all RALF users, I found RALF a little hard to understand at first; but after a little experience with it, it became incomprehensible. I published some very painfully acquired knowledge in #18, p. 16; now I find it could all have been prevented if I'd read Lars Palmer's note in #15, p. 4. Another example: RH explained how to write OS/8 system handlers in #11, pp. 1-2. Some of the same ground was covered by Steve Ligett in #16, pp. 22-3, and Ian Templeton, #16, pp. 10-11. I don't know if they had read #11, but suspect that they hadn't.

Virtually all of Stan Rabinowitz's material is still relevant, and much of it is very important. I scored about 50% on his OS/8 quiz (#15, p. 18), and learned some very useful things.

12-bit SIG may not quite be ready for a "5 and 10 years ago" column, but in April 1972 RH noted that "more and more it is becoming evident how much a problem [medium compatibility is.] So far DECUS is not in a position to convert program materials . . ." Nothing much has changed, except that there are now more media. On a more encouraging note, in April 1974 he **announced** that "initial design work on . . . a full scale relocatable MACRO assembler and loader is under way."

I'd like to see a "best-of-12-bit-SIG" compiled and distributed a) to new SIG members, and b) with every copy of the Software Support Manual, but I suppose there's about as much chance of that happening as, say, DEC adopting DIRECT v5G or fixing the bugs in FORLIB SYNC . . .

OS/8 FORTRAN IV QUIZ

1. How does OS/8 FORTRAN IV treat the statement:

```
DO 100 I = 1.9
```

- a) Gives a misleading diagnostic message.
- b) Compiles it normally.
- c) Gives the diagnostic message "DO" (syntax error in DO).
- d) Treats the decimal point as if it were a comma.

2. Which of the following can significantly increase execution speed of an OS/8 FORTRAN IV program? (more than one may be true)

- a) Use integer arithmetic wherever possible.
- b) Avoid multiply-subscripted arrays.
- c) Use the /N and /Q options.
- d) Add an EAE to your PDP-8I or PDP-12.
- e) In a critical loop, code in-line rather than calling a short subroutine.

3. What happens if you call SIN(3), that is with an integer rather than a real argument?

- a) The compiler catches the error.
- b) The loader catches the error.
- c) The run-time system catches the error.
- d) The smart run-time system calls FLOAT and converts the argument to a real, and the correct result is returned.
- e) Nobody catches the error or performs any conversion, but the correct result is returned anyway.
- f) The value 8.85E315 is returned, because the argument wasn't normalized.

4. You have written a complex FFT subroutine that explicitly uses type COMPLEX variables and arrays. On which of the following systems will it run without producing an error message? On which will it run correctly?

- a) basic PDP-8.
- b) PDP-8 with EAE.
- c) PDP-8 with single-precision FPP option.
- d) PDP-8 with double-precision FPP option.

5. Name two cutesy features of the PASS3 (listing) program that will probably be enjoyed by high-school students.

6. What will the following function return the fifth time it is called, a) according to ANSI standard FORTRAN, b) under OS/8 FORTRAN IV? (Caution: tricky!)

```
FUNCTION NTH(IGNORD)
DATA N/0/
N = N + 1
NTH = N
RETURN
END
```

7. Where in the OS/8 manual can you find a convenient reference table for use with CPUT and CGET, giving the ASCII character code truncated to 6 bits and converted to decimal?

8. You have written a SECT8 module in RALF. About a dozen locations from the start of the page, you have the following code:

```

RTL6, 0
    CLL RTL
    RTL
    RTL
    JMP% RTL6

```

This subroutine is called several times via a "JMS RTL6" from the main routine. The module usually works perfectly. In a given program, if it works at all, it always works. Whether or not it works seems to depend on what other modules are loaded with it, although it does not reference any other modules. Why?

9. What additional requirements must an OS/8 device handler satisfy, above and beyond those described in the OS/8 software support manual and BUILD, if it is to work properly under FORTRAN IV?

10. The FRTS command decoder lines

```

*AFILE/8
*BFILE</9

```

define AFILE as an input file and BFILE as an output file. True or false?

Answers:

1) b. This is a perfectly correct FORTRAN statement, signifying that a real variable named "D0100I" is to be given the value 1.9.

2) b, c, e. Integer arithmetic is no help because all arithmetic in OS/8 FORTRAN IV is done in floating point. Avoiding multiple subscripting can help because each multiple subscript requires a calculation involving a chain of (floating-point) additions and multiplications; naturally, no benefit will accrue if you simply replace them with explicit additions and multiplications in your own code, but often this can be avoided by tricky use of "EQUIVALENCE," etc. The /N and /Q options produce noticeable, but rarely dramatic increases of speed. FORTRAN IV EAE support is strictly for mode B, which older EAE's lack, so the EAE will help only on PDP8E's and later. Subroutine calls are extraordinarily complex and can make a noticeable difference in a tight loop.

3) e. In OS/8 FORTRAN IV there is virtually no difference between integers and reals. Integers are stored in the standard, 3-word, normalized floating-point format; the integer 3 and the real 3.0 have identical internal representations. As far as I know, the only way in which the compiler differs in its treatment of integers and reals is that it calls JSA #FIX before storing a real value in an integer variable.

This feature of FORTRAN IV has both good and bad aspects. The obvious bad aspect is that all arithmetic is done in floating point, at floating point speeds. (This includes DO-loop indexing calculations and array subscript calculations! Well, you wouldn't want DO loop variables to be restricted to the range -2048 to +2047, would you?) The good aspects are enormous simplifications in all phases of the system.

It is automatically true that any function, in the library or user-written, that expects a real-valued argument, will accept an integer argument and do the right thing. It is also quite OK to print an integer under an F format, or a real under an I format (it will round to the nearest integer!). It is possible, through argument-passing or other means, for an integer to acquire a real value; this is not quite so harmless, but the effects can be logically predicted (e.g. I.LF.3 and I.GT.2 could simultaneously occur). Very stern self-discipline is required to maintain an awareness of the proper FORTRAN conventions regarding integers and reals in the permissive OS/8 environment.

4. d; a, b, d. On systems a, b, and c it will produce the message "CAUTION--NO DP," but it will nevertheless run correctly on systems a and b. The caution message is produced because the complex arithmetic routines use STARTE mode loads and stores; loads and stores, but no arithmetic, are included in the FPP simulator.
5. a) the nature of the error message produced when a keyword has been misspelled; b) the manner in which the date is printed on certain days of the year. (And while we're on the topic-- I assume everybody knows the response to the CCL command .MAKE LOVE?)
6. a) According to ANSI standard FORTRAN, the value of N, and hence the value returned by NTH, becomes undefined on execution of the RETURN statement (X3.9, 10.2.6). That is, the first call returns 1, but subsequent calls return undefined values.
b) If the program contains no overlays, the value of N will be preserved between calls, and the fifth call will return 5. If NTH is part of an overlay structure, and a call is made a subroutine in a different overlay on the same level, on the next call to NTH it will be recalled from mass storage and N will become zero again.

MORAL: if your program has slowly grown and has just gotten too big for core, eyeball the routines before putting them into an overlay structure.
7. p. 8-133, table 8-20 of the plotter routines.
8. SECT8 routines can load anywhere; depending on the size and number and sequence of other modules, your routine might or might not load into PAGE 0 of some field. If it does, RTL6 is in an autoindex location.

9. Shades of question 6! Handlers that run under FORTRAN may not contain any internal flags, counts, etc. that is to be preserved between calls. FRTS only has room in field 0 for one handler at a time, so the handlers are loaded into high core and copied down as needed. If handler A is resident and B is needed, A is simply wiped out; its internal state is not preserved (e.g. by copying it back up to high core). The next time A is required, a fresh copy will be made and it will revert to its initial state. Thus, for example, a line-printer handler that attempts to remember how many lines have been printed since the last top-of-form will not run properly under FORTRAN IV.
10. False. They specify that AFILE already exists while BFILE is to be newly created. AFILE may be written into, but its length is fixed and may not be extended. BFILE may be read from, but its initial contents are undefined.

SCORING:

- less than 2: DEC salesman
- 2-3: ~~DEC~~ OS/8 FORTRAN IV maintainer
- 4-5: FORTRAN programmer
- 6-7: OS/8 FORTRAN IV programmer
- 8-9: Programmer
- 10: Your mind has been permanently damaged through close association with OS/8 FORTRAN IV.

Dear Bob,

As an old PDP fan I have ventured into almost every art of programming the 12-Bit computers. From DECTrieve to Propofop and the disk monitor to PS-FOCAL. Lately I got acquainted with DIBOL, and I must say I do like it very much. With the low memory prices nowadays, one can afford to use a comfortable compiler. But of course, this is not the reason I am writing. I think I should tell the SIG members of what I accomplished with DIBOL. It all started with our sales force, and their cumbersome task of quotation writing for more than 1000 different products. Management found that a computer manufacturer should be able to automate at least some of these tasks. To make a long story short, I found myself confronted with the job of producing a computerized automatic quote writing system (AGS). Together with 2 programmers from an outside software house we solved the problem in a (what I think) very elegant way within 8 months time, without neglecting our regular duties. Now, let me give a compressed overview of the system:

There are 4 major parts: the commodity file system, the customer file system, the parameter files and the quote system. The commodity file contains a base record and up to 98 description text records for each article filed. The base record contains the part number, a product line code, the sales prices, the discount level if applicable, the maintenance and the installation charge. Two utility programs, one for the upkeep, another for printout with various formats and prices.

The customer file contains a customer number, 8 alphanumeric characters, 4 lines, 60 characters each for the customer's name, 40 characters each for the street and city, 5 for the zip code and 2 numerical digits for arbitrary coding. There are utility programs for upkeep and for the printing of address labels. Search criteria are the customer number, the numerical code and an alphanumeric search over the first 2 lines of each entry.

The 2 parameter files contain up to 20 different price factors for the modification of sales and service prices, the percentage of additional value taxes, the currency i.e., dollar, pound, lira, deutschmarks etc. and a minimum price limit which may be applied to an installation charge of x percent of the sales price but not below the given minimum etc.

The second parameter file contains the office address and tel. number, the office account number, a crossreference between all salesmen and their badge numbers and the automatic running number from 0 ... 999 (restarting) being applied to the project number of each quote.

The quote system starts with the acquisition of customer, remarks, contract conditions (OEM, EU, BVB, etc.) Discount levels and percentages, installation charges or not, grand total or not, surcharges or deductions, part numbers etc., it allows text modifications, additions, text or part insertions and updates to any degree. Upon request every or any line may be displayed on the screen and is available for changes. The quote may then go to the spooler or be printed right away. It also is recorded on the quote file for later reprint, investigation or as input to other quotes. A short version (without descriptive texts) goes on to the office log file. Utility programs allow upkeep, printouts and statical investigation of the log file e.g. sums over a certain period, orders only etc. Cross references between customer order numbers and internal order numbers are available. Each quote bears a project number consisting of the salesman's account and badge number and a running number between 0 and 999 which is automatically stepped forward by the computer and restarting when the limit is reached. Each quote also bears the customer address, the issuing office address, date and salesman's name (see attached sample), it also prints the applicable terms and conditions, the approx. delivery time and validity time of the quote.

-2-

The hardware is a PDP8/A with 16 kilowords of core, 2 floppy disks, 2 RK05 disks, VT50 and LA180. Parts of the system may run on smaller hardware configurations, and if the commodity file can fit onto 1 floppy (approx. 400 parts with descriptions) the whole system may get away with one RK05 disk. 4 floppy drives are not recommended since they are too slow. We run it such, that each salesman comes to the system with his own customer and his own quote floppy, everything else is kept on the large disks. Floppies are also used as update media to all offices e.g. when the commodity file is changed or parameter files etc. Corresponding floppies are sent to each office from the central manager who only can change parameter and commodity files.

We are now in the second month test phase in order to find out what errors have to be fixed or improvements could be made. From the start, opinions were parted, but with the time, more and more intusiasm is building up and large quotes, which used to take up to a full day of a secretaries time may be done within 10 to 15 minutes without any price calculation errors. Salesmen may call from the customer they visit and give their secretary all the info she needs to fill the quote acquisition form with which she goes to the terminal to produce the quote which the salesman finds ready to be signed on his desk when he returns from his visit. When a customer visits the office, hears all about a proposed computersystem and asks for the price, he is all surprised when he sees the computer prints his own quote within minutes.

Also salesmen may print their own price lists in parts or as a whole from the commodity file with respective OEM, end-user or special contract prices. They may have a printout of a descriptive text in order to find out what parts belong to a system or article.

I am pretty shure, that DEC will sell the software without any warranty or maintenance obligation after the test phase for a nominal fee to interested customers. Maintenance is releatively easy for DIBOL is written in plain English and a program is transparent without comments. The software house, which took part in the development work is using the system also with great satisfaction. This may show you, that also within DEC the PDP-8 is ranking high.

regards,

Rudi Stange
Sales Support
Digital Equipment GmbH
Wallensteinplatz 2
8000 Munich 40

NORTHWESTERN UNIVERSITY
EVANSTON, ILLINOIS 60201

DEPARTMENT OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE

THE TECHNOLOGICAL INSTITUTE
(312) 492-5410

December 22, 1977

Mr. Robert Hassinger
Liberty Mutual Research Center
17 Franklin Road
Hopkinton, MA 01748

Dear Bob:

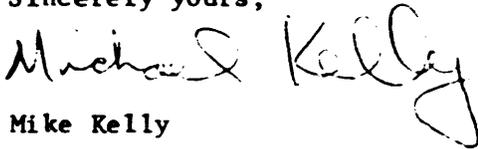
Enclosed please find an abstract of the magtape file system we are writing for OS/8. We would appreciate it if you would print it in the next 12 Bit SIG Newsletter, so others can become familiar with our goals and philosophy.

If there is sufficient interest, we may hold a "birds of a feather" session at the Spring Decus. We would like interested parties to read our abstract, and either bring their ideas to the Spring Symposium, or write us directly. The address is:

Mike Kelly
Northwestern University
Computer Science Research Laboratory
2145 Sheridan Road
Evanston, IL 60201

Thanks for your help. Hope to see everyone at the Spring Decus here in Chicago.

Sincerely yours,


Mike Kelly

MTF - Magtape File System for OS/8 Abstract

In many installations where file storage is a premium, an easy method of rolling off seldom used files is needed. Until now, the best method was to use DECTape, which, while acceptable for small installations, is quickly outgrown by large, production-oriented systems. The solution is magtape. A typical 2400 foot reel of magtape can hold over 20,000 OS/8 blocks. Access time is slow, but rolled off files are usually inactive anyway.

The problem with magtape has been OS/8's marginal support. No attempt has been made to support labelled tapes of any kind, but this is understandable due to the two page handler limit imposed by OS/8.

We considered two methods of implementing magtape into OS/8. The most obvious method is to add more sophisticated label processing to MCP/IP (MCP/IP currently supports cassette labels). The advantages of this method are that it requires minimal modification to the OS/8 system itself (i.e. rewriting the tape handler and MCP/IP), and also allows users with as little as 8K to access magtape. However, this method was rejected because:

- a) This would often require copying files from tape to disk to allow convenient access by other OS/8 CUSPs. A major advantage of tape is its size. Many files which are placed on a tape cannot be easily copied to a disk because of space restrictions.
- b) Most systems large enough to require magtape have at least 16K, the current MTF requirement.
- c) Automatic support for GET and PUT (Wrege and McIntyre, West Virginia University), the most common backup/restore programs, as well as support for any user written programs.
- d) The current tape handler cannot be rewritten within the framework of OS/8, since the new handler would require much more than 2 pages. The current handler uses a non-standard (six bit) format, which is usually unreadable by other installations. So, while OS/8 does support magtape, the support is non-standard. Rewriting MCP/IP would require adding

**MTF - Magtape File System for OS/8
Abstract**

a magtape handler directly into MCPIP, thus allowing only MCPIP to access standard format tapes. Other OS/8 CUSPs would have to use the OS/8 handler, introducing two formats: the standard format, and the OS/8 format.

The method we have adopted solves all the above problems. We have decided to implement magtape as an OS/8 device. In order to accomplish this, we must avoid the two page handler limit imposed by OS/8. This is done by running OS/8 in the background, and RTS/8 in the foreground (using the OS/8 support task, OS8SUP). OS8SUP will trap all OS/8 handler calls and pass them to RTS/8 tasks. This allows us to write our tape driver and label processor as RTS/8 tasks (with no size limitations), and have OS8SUP pass all OS/8 references to any of the devices MTA0-MTA7 to an OS/8-RTS/8 interface task. This allows most OS/8 CUSPs to access magtape with no modifications.

The disadvantages of this solution are:

- a) A reduction in OS/8 response time, due to RTS/8 overhead.
- b) No support for any OS/8 program which is interrupt driven (since RTS/8 uses the interrupts) or programs which do their own I/O (except I/O to TTY, LPT, and High speed paper tape), including the following CUSPs:

BUILD (does own I/O)
 BOOT (does own I/O)
 PIP10 (does own I/O)
 INDUSTRIAL BASIC (interrupt driven)
 BASIC (Interrupt driven)
 FORTRAN LAB FUNCTIONS (do own I/O)
 FORTRAN IV (interrupt driven)

- c) No automatic support for BATCH access to magtape files, but slight modifications could add this support.

MTF introduces a new device concept to OS/8 -- file-structured, non-directory devices.

MTF - Magtape File System for OS/8 User Interface

MTF allows the user to access any file on a magtape regardless of current tape position. Utilities such as CAMP are no longer required to position the tape (although CAMP is supported). Positioning is done via labels. These "labels" are similar to directory entries on directory devices. The difference is in their position. Instead of collecting all the labels together in one directory file, each file is preceded by its directory entry. American National Standards Institute (ANSI) has defined a standard for magnetic tape labels in their document X3.27-1969 published on June 20, 1969. This standard is now supported by MTF. For a complete description of the label format, see appendix A. The new 1977 ANSI standard (which is not yet published) may be supported by MTF in the near future. However, the exact format of the label does not concern the user, since MTF makes labels invisible to both OS/8 and the user.

To illustrate the MTF flow of control, we will assume that some OS/8 CUSP (PIP, for example) has been instructed to write a file to tape. The command decoder line might look like

```
*MT.1:<FILE.TX
```

PIP invokes MTF through the OS/8 Support Task, OS8SUP. MTF will then call the Command String Interpreter (similar to the OS/8 Command Decoder) to obtain a file assignment. This file name is passed to the label processor. A label will be constructed for the file, and written after the last file on the tape, effectively entering a new file. Once the tape is positioned after this label, control returns to PIP. PIP will now call MTF each time an I/O transfer is to be made. Since MTF knows it already has a file name, it will simply call the Magtape Driver (MTA) to perform the I/O, and return to OS/8. When OS/8 calls with a CLOSE operation, an EOF label is written on the tape effectively closing the output file. Figure 1 graphically illustrates the flow of control, and the role each software component plays in the system. Note that MTF is itself a component, although it is the only component visible to the user, and thus it is the name for the entire system. All other components are controlled by MTF, and are effectively a part of MTF itself.

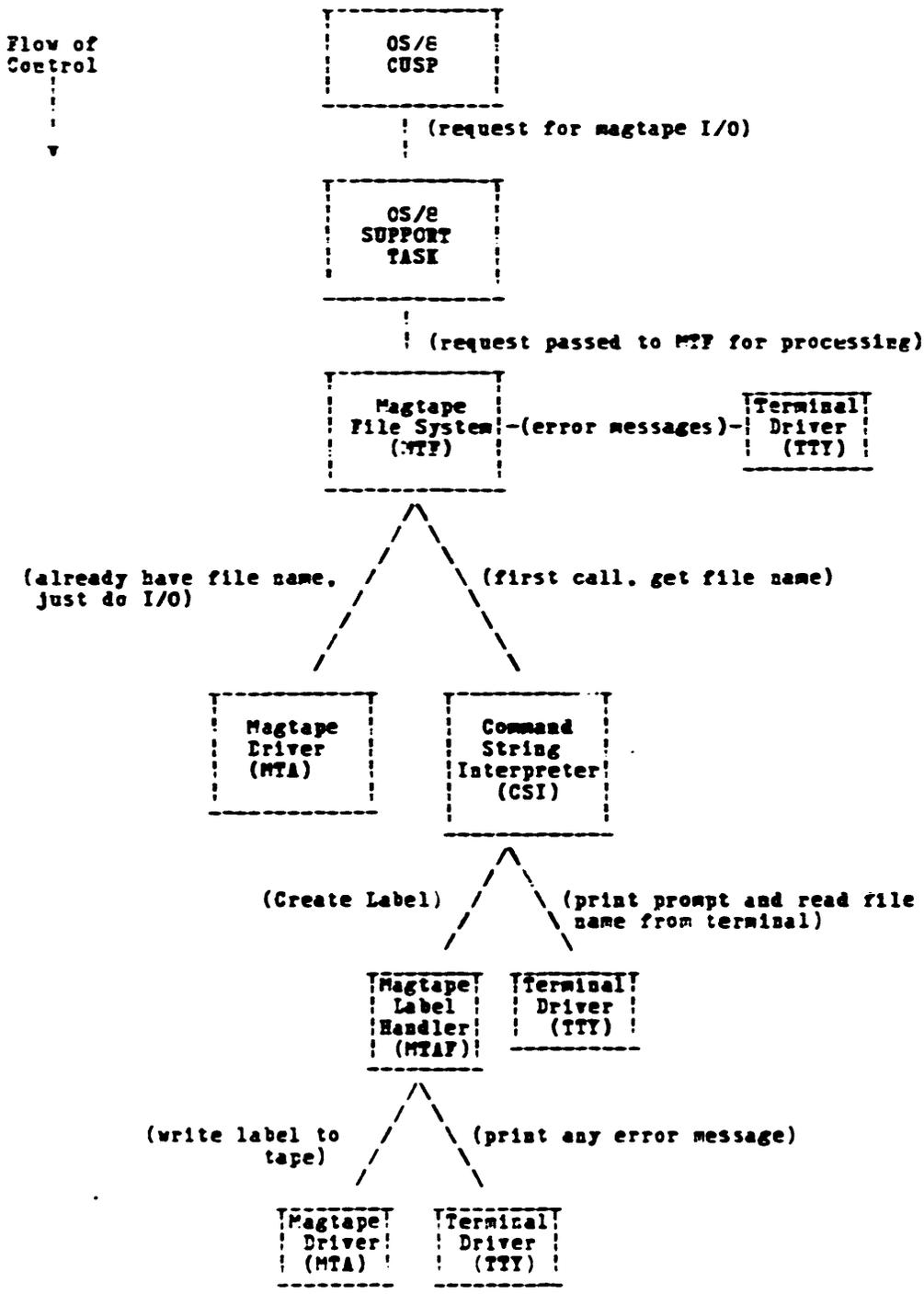


Figure 1 - MTF Flow Diagram

MTF - Magtape File System for OS/8 Software Specifications

While designing the MTF system, several goals and constraints were kept in mind. These are briefly described below.

- * Execution speed was optimized as much as possible, but OS/8 support speed will compromise most of the RTS/8 overhead (neglecting RT8/A memory management), so this goal was given a low priority.
- * Compatibility with accepted ANSI standards, and with the rest of the DEC world was a major design goal. As a result, we have stand alone programs to read and write both ANSI standard and DOS-11 format tapes. We hope to eventually integrate these, as well as several other, formats into the MTF system.
- * Software portability was also a major design goal. Thus, all software, with the obvious exception of the actual tape handler, was written to be controller independent. That is, the only modifications necessary to drive a non-TC58 controller are to the tape driver. The driver must be either modified or rewritten to our software specifications.
- * Backup/Restore operations have been kept in mind. Our system requires routine backup of 20 megabytes (yes, we do have 20 mb on one PDP-8!), which would require 8 RK05 packs (at 2.5 mb each), or 75 DFCTapes (at a mere 0.2 mb each). One 2400 foot reel of tape can hold 8.1 mb.
- * Our system is composed of a network of mini-computers, including a VARIAN 73, a PDP-11/40, several LSI-11's, several PDP-8's, an Intel 8080, and a high speed link to the main university computer, a CDC 6600. All of the mini-computers in our network share one magtape drive, so system independence is a must. We needed a standard which could be implemented across the board, and also be read by the main university computer. The university system has not fully evolved their tape system, so we needed something which would always be standard. The obvious choice was the ANSI standard, which is supported both by the 6600, and by RSX-11/M, the operating system for the 11/40 (it appears DEC is beginning to understand standards are important). Not only does ANSI allow us to

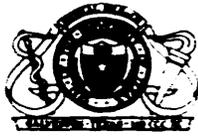
**MTF - Magtape File System for OS/8
Software Specifications**

transfer programs among the different systems we have contact with (we recently read a PDP-11 tape on our PDP-8), but also allows us to transfer files between our network and the main university computer via tape. In the event of a bad system crash, the system could conceivably be restored via the high speed link, since our tapes are readable by the 6600.

- * We have also undertaken the goal of media management, and ANSI labels are in keeping with this goal. Our general medium management package (which we have modified from the original MEDIAI/MEDIAO package written by McIntyre at West Virginia University) will support magtapes, as well as all other DEC file structured devices. This allows us to produce a listing of all files currently preserved, and where they are preserved.

MTF is supported by the Northwestern University Computer Science Research Laboratory. Suggestions for improvements and comments on performance should be sent to:

**Mike Kelly
Northwestern University
Computer Science Research Laboratory
2145 Sheridan Road
Evanston, IL 60201**



THE UNIVERSITY OF TEXAS MEDICAL BRANCH
GALVESTON, TEXAS 77550

November 21, 1977

Mr. Robert Hassinger
Liberty Mutual Research Center
71 Franklin Rd.
Hopkinton, Massachusetts 01748

Dear Bob:

Several of us are having problems running Camp and MCPIP on PDP8/E computers equipped with a magtape interface and controller supplied by Datum Inc. This hardware is suppose to look to the PDP8/E like it is D.E.C.'s TM8E controller. It is possible for us to run the TM8E diagnostics supplied by D.E.C. but Camp and MCPIP fail for various reasons. Datum thinks Camp and MCPIP are at fault but my software contacts at D.E.C. say that their programs work properly with their own TM8E controller. We have not been able to find any PDP8/E users running a D.E.C. supplied TM8E controller under OS/8. It would be helpful to us if we could hear from someone successfully running Camp or MCPIP with D.E.C. magtape hardware or anyone else's hardware.

My address is: Dr. James L. Rae
Department of Ophthalmology
University of Texas Medical Branch
Galveston, Texas 77550
(713) 765-2176

We would appreciate any help we can get for this problem.

Sincerely,

James L. Rae

James L. Rae, Ph.D.
Associate Professor
Department of Ophthalmology,
Physiology and Biophysics

JLR:bb

/ OS/8 DIRECT V5G

PAL8-V9H 12/15/77 PAGE 13

4000 *4000

14000 0000 TYPE, 0

// UNDER ETOS, MAKE THE FOLLOWING .ODT PATCH:

14001 6046 TLS
 14002 7200 CLA
 14003 5600 JMP I TYPE

4001 *.-3

14001 3321 DCA GTSXBT
 14002 1377 TAD (217 /~O ?
 14003 4224 JMS CTYPE /CHECK FOR IT.
 14004 3260 DCA ECHO

ETC...

Abbey/Portsmouth R.I. 02871
 15 December 1977

Dear Bob:

The above indicates a patch worth making to DIRECT (version 5G) when it is run under ETOS. It's easily done and takes out a lot of unnecessary IOT codes. ETOS itself handles Ctrl/O and the like.

Some day I plan to have a second look at the internal sort, currently a 'bubble' sort and rather slow.

I can duplicate paper tape (PCBE punch here) or Dectape copies of this version. The features are worth having.

I no longer receive SPR's or Software News. I hear rumours of a charge for these, some sort of 'software maintenance contract'; but even this I don't hear about from DEC!

One user, getting a royal run-around from someone in Maynard over the phone, expressed exasperation. He was sharply rebuked: "Don't you realize we're a BILLION DOLLAR COMPANY now?"

Maybe Communism isn't so bad after all...

I reformatted the DIRECT.5G source to print on 80-column paper. The copyright etc. notices are still there.

Yours for the revolution,

G. Chase

(Rev. Don) Geoffrey Chase, OSB

**GENERAL LATEX AND CHEMICAL CORPORATION (OF OHIO)**

P. O. BOX 498 • ASHLAND • OHIO 44805
419-289-2727

December 18, 1977

Robert Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, Mass. 01748

Dear Bob:

The enclosed abstracts are on the way to DECUS, but if past performance is any criterion, DECUS catalog update will be 12-15 months coming.

ALPHA is primarily updated for V3D dates, but also adds DECSYSTEM-8 parameter block information. However, SPY, FUTIL, DX, etc., can set up a meaningful volume identification which will be picked up by ALPHA without a full fledged parameter block. 8BIT ASCII terminated by 232 (control-Z) in absolute block 15 is all that's required.

LTFRMT represents extensive recoding of DEC's MARK12. I have enclosed the listing and an LTFRMT.BN papertape -- I think you (sometimes) use a PDP-12. All CPU halts have been removed and are replaced by console error message type-out. Unsuccessful formatting is also indicated by a console message showing what type of error condition has occurred. Directory zeroing, automatic volume sequencing, and DECSYSTEM-8 parameter block filling are now a part of LTFRMT. All the needed information for any length LINCTAPE is put into the respective blocks on the tape. This tends to reduce the problem of PIP zeroing non-standard length LINCTAPES, but does not eliminate the problem. Basically, just don't use the PIP zero. The DECSYSTEM-8 zero will work better anyway. Incidentally, not only can long LINCTAPES be created, but so can SHORT ones.

I have successfully used tapes marked with up to 1007 OS/8 blocks, however, 1007 blocks is a very, very touchy marking job. Using both LINCTAPES and DECTAPES purchased in 1971, 1972, 1974, 1976, and 1977, I find a good 'practical' maximum of 967 OS/8 block (of either 128 or 129 words) will format every time with 3-5 wraps around the take-up reel. In other words, no special handling of the tape on the drive. The write-up has a table appendix indicating 17 formats and their associated variables. I have also marked 256(10) - 1024(10) word blocks, but don't have any use for this format (yet).

I think LTFRMT will be a quantum jump improvement for PDP-12 users under OS/8. It provides the combined functions of formatting, zeroing, and parameter setting at the same time, and in no longer time than the original MARK12 program. As a matter of fact, DTFRMT could stand the same type of rewrite!!

Sincerely,

H. S. Hopkins Jr.

This is a tape PDP-12 users may be interested in also.....

These are the PDP-12 MAINDECS translated from DIAL to OS/8, with a starting location of 7604 (HLT). There is plenty of room to add more MAINDECS, such as the FPP-12 etc. Note that this is a 900 block SYSTEM tape.

Fielder

17-DEC-77 FILENAME SORT
 PARAMETER BLOCK CREATED 20-NOV-77
 LT VOLUME 1
 0955 BLK LINCTAPE VOL 001
 TAPEMARK Q ON 11/20/77
 OS/8 PDP-12 MAINDECS

FILENAME	LEN	SAVE DATE	FILENAME	LEN	SAVE DATE
ADRSHI. MD	4	02-NOV-77	JMPSEL. MD	2	02-NOV-77
ADRSLO. MD	3	02-NOV-77	KF12B . MD	16	02-NOV-77
ADRS12. MD	2	02-NOV-77	KW12A . MD	17	02-NOV-77
ADTST . MD	3	02-NOV-77	KW12BC. MD	5	02-NOV-77
ALPHA . HL	3	20-NOV-77	LP08 . MD	14	02-NOV-77
ALPHA . SV	10	20-NOV-77	LS8E64. MD	15	02-NOV-77
BINLDR. MD	3	02-NOV-77	LS8E96. MD	15	02-NOV-77
CALCOM. MD	8	02-NOV-77	LTFRMT. HL	2	02-NOV-77
CBH . MD	4	02-NOV-77	LTFRMT. SV	12	02-NOV-77
CBL . MD	3	02-NOV-77	MEMCT . MD	10	02-NOV-77
CB12 . MD	3	02-NOV-77	MEMDA . MD	2	02-NOV-77
CCL . SV	17	17-NOV-77	ONOFF . MD	3	02-NOV-77
CPTST1. MD	4	02-NOV-77	PWRFAL. MD	3	02-NOV-77
CPTST2. MD	17	02-NOV-77	RANISZ. MD	5	02-NOV-77
CPTST3. MD	13	02-NOV-77	RANJMP. MD	4	02-NOV-77
DB12 . MD	5	02-NOV-77	RELAY . MD	4	02-NOV-77
DC02E . MD	6	02-NOV-77	RF08DD. MD	11	02-NOV-77
DC02F . MD	9	02-NOV-77	RF08MD. MD	5	02-NOV-77
DIRECT. SV	7	07-JUN-77	RIMLOW. MD	3	02-NOV-77
DP12A . MD	12	02-NOV-77	RK8EDA. MD	9	02-NOV-77
DSPTST. MD	4	02-NOV-77	RK8EDC. MD	15	02-NOV-77
EAE3A . MD	13	02-NOV-77	RK8EDL. MD	16	02-NOV-77
EAE3B . MD	12	02-NOV-77	TAPEDA. MD	4	02-NOV-77
EXTAT . MD	7	02-NOV-77	TC12EX. MD	8	02-NOV-77
EXTMC . MD	6	02-NOV-77	TC12F . MD	8	02-NOV-77
INDEX . WU	12	02-NOV-77	TC12I . MD	17	02-NOV-77
INST1 . MD	11	02-NOV-77	TC12II. MD	13	02-NOV-77
INST2 . MD	10	02-NOV-77	TTY1 . MD	7	02-NOV-77
INST2A. MD	12	02-NOV-77	TTY2 . MD	11	02-NOV-77
JMPJMS. MD	4	02-NOV-77	XTMC12. MD	13	02-NOV-77

OS/12 SYSTEM DEVICE
 2 DIRECTORY SEGMENTS USED
 60 ACTIVE FILES IN 496 BLOCKS
 404 EMPTY BLOCKS

A B S T R A C T

A L P H A

(Sort OS/8 Directories)

ALPHA is an OS/8 utility program to sort directories on any one of the four keys contained: filename, extension, creation date, or starting block number. ALPHA V03 is an extensive revision to the original DECUS 8-825 submission; primarily to update for the extended date of OS/8 V3D, with additional enhancement provided in the form of a heading print of the sort method, and printout of the DECSYSTEM-8 parameter block information. Operating systems provided for are PS/8, OS/8, OS/12, DECSYSTEM-8, and OS/78. Extensive checking of the parameter block information prevents 'runaway' garbage printing in the case DECSYSTEM-8 parameter blocks are not being used.

ALPHA can be chain called; CCL41.PA provides the keyword AL(PHA) monitor command in addition to other PDP-12 oriented features.

The "1" option has been modified to automatically set up the equivalent to a command decoder line of *LPT:<SYS:;DSK:(BEHN)=2, giving easy access to the devices most often used under OS/8. Calls to print the same directory multiple times are totally inhibited so that only one copy of each requested device will be printed per pass.

The library submission tapes (LINCTAPE and DECTAPE) contain the following files: ALPHA.PA, ALPHA.BN, ALPHA.SV, ALPHA.WU, ALPHA.HL. CCL41.PA is also included for guidance in setting up a call to ALPHA via CCL.

A B S T R A C T

L T F R M T

(LINCTAPE Formatting Program)

LTFRMT is a significantly enhanced replacement for all current versions of MARK12, and can be used under either DIAL or OS/8.

Formats provided are the standard 1, B, P and X; plus the addition of a new Q option allowing an operator choice of both the number of blocks and the words/block to be marked. (This feature has a primary use of creating long LINCTAPES for use under OS/8.)

Under OS/8, and only through a chain call, LTFRMT also provides:

- 1) Zeroing of OS/8 directories (either sys or nonsys)
- 2) Automatic sequencing of volume numbers (001-999)
- 3) Set up a DECSYSTEM-8 parameter block with a dummy identification frame
- 4) Provides for extended date of OS/8 V3D
- 5) Removes all possible CPU halts
- 6) Gives console messages if formatting is unsuccessful

LTFRMT has been designed as a true "system" program in the sense that only a hardware failure will cause either a CPU halt or the need to restart the system from the processor switch registers.

A write-up file provides extensive documentation for LTFRMT. The LINCTAPE library submission contains the following files: LTFRMT.12, LTFRMT.BN, LTFRMT.SV, LTFRMT.WU, and LTFRMT.HL. CCL41.PA contains table modifications to run LTFRMT.SV by a monitor call (MARK) which is required to take full advantage of the OS/8 features of LTFRMT. Other additional PDP-12 oriented enhancements are also included in CCL41.PA.

Under either OS/8 or DIAL straight monitor calls, LTFRMT provides identical operation to MARK12 plus the addition of the "Q" option.

NOTE...LTFRMT.12 (source file) must be assembled under PAL12, because of the LINC mode coding. DIAL assembly is not possible because of differing techniques of handling the "TEXT" pseudo-op.

 A B S T R A C T

C C L 4 1 . P A

CCL enhancements for the PDP-12

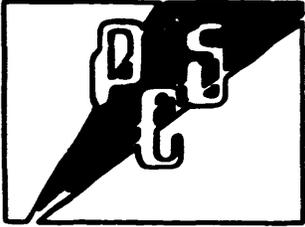
CCL41.PA is a modification to CCL.PA (Edit 39) provided with OS/8 V3. All Digital Software News 'fixes', etc., have been included plus the following PDP-12 oriented enhancements:

Keyword	Calls	Operation
AL(PHA)	ALPHA.SV	Sort OS/8 Directories
MARK	LTFRMT.SV	Format LINCTAPES
DX	DX.SV	Directory display on VR12/VR14 CRT
EDIT	SCROLL.SV	Edit existing file on device DSK
CREATE	SCROLL.SV	Create a file on device DSK

Modification to other CCL Tables:

DIRECT..... CCL default output device is TV
 ALPHA..... CCL default output device is TV
 RESORC..... CCL default output device is TV, and includes /E
 MAP..... CCL default output device is LPT
 COPY..... CCL message removed, /L removed
 RENAME..... CCL /L removed
 EDIT..... CCL calls SCROLL.SV to edit an existing file
 on (the user assigned) device DSK
 CREATE..... CCL calls SCROLL.SV to create and existing file
 on (the user assigned) device DSK

Note that compatibility of CCL.PA versions distributed with OS/8 V3C and after may not be good with CCL41.PA. Source patching of your version of CCL is recommended, and CCL41.PA has been included only as an example. CCL41.PA has set CCLVER and CCLTAB to "H".



PROCESS CONTROL SYSTEMS, INC.

18130 S. Thorapple Lane

• New Berlin, Wisconsin 53151

• (414) 782-3945

HARDWARE CONSULTATION

November 1, 1977

SOFTWARE DESIGN

PROCESS CONTROL SYSTEM DESIGN

Mr. Robert Hassinger
12 Bit SIG
c/o DECUS
1146 Main Street
Maynard, MA 01754

Dear Bob:

In this letter I will describe OS/8 Industrial Basic by saying it is identical to OS/8 Basic with the addition of these statements and functions: (Taken from "Industrial Basic Language Reference Manual", DEC-S8-OS8BA-B-D)

CONTACT V THEN N Define the line number (N) to be scheduled when a contact interrupt occurs on point (V).

COUNTER V THEN N Define the line number (N) to be scheduled when the counter point (V) reaches zero.

TIMER V THEN N Define the line number (N) to be scheduled when time V has elapsed. Continue to cycle until timer disabled.

CONSOLE V THEN N Define the line number (N) to be scheduled when the character defined by V is typed on the terminal.

CONTROL P THEN N Define the line number (N) to be scheduled when INDUSTRIAL 14 P interrupts.

DISMISS Exit from the user process interrupt mode - resume main line code.

NOTE: Fulfillment of the conditions in the above statements cause a UPIR (User Process Interrupt Routine) to be executed. A UPIR is like a subroutine, but it is called by an external event, not by a programmed event. When a UPIR is scheduled, whatever was happening is suspended, the UPIR executes until a DISMISS statement is encountered, then the mainline or background program picks up from the point it was interrupted.

BIS(L,R) This function converts the expressions L and R to two 12 bit PDP-8 words. It then computes a bitwise INCLUSIVE OR and returns the resultant 12 bit word as an integer.

BIC(L,R) This function converts the expressions L and R to two 12 bit PDP-8 words. It then complements L and AND'S it with R. The result is returned as an integer.

OCT(X\$) This function scans the string X\$, ignoring blanks, and computes a 12 bit octal number

con't

represented by the string X\$.

BIN(X\$) This function scans the string X\$, ignoring blanks, and computes a 12 bit binary number represented by the string X\$.

OCS\$(X) This function converts the expression X to a 12 bit integer. The integer is then converted to a 4 character string representing the octal value of the integer.

BNS\$(X) This function converts the expression X to a 12 bit integer. The integer is then converted to a 12 character string representing the binary value of the integer.

ANI(C,G) This function reads analog input point C at gain G and returns the value in volts.

ANO(C,V) This function loads the analog output channel C with the value V.

CLK(X) If the value of X is zero or negative, this function returns the time of day clock in seconds. If the value of X is positive, the time of day clock is set to the value of X.

CNI(P) This function returns the number of counts remaining until zero in counter P.

CNO(P,V) This function loads the counter P so that V items are counted before the counter interrupts (overflows).

RDI(P,N) This function returns the value of points P through P+N-1.

SDO(P,N,V) This function loads the value V, in binary, into points P through P+N-1 (right justified).

RDO(P,N) This function returns the results of the last data sent to points P through P+N-1.

LNE(X) Returns the channel number of the contact module that caused the interrupt.

STA(X) Returns the state of the contact associated with the channel that scheduled the UPIR.

CNT(X) Returns the number of identical schedule requests for the same UPIR.

RDB(P,A) RDB reads I/O address A of controller P and returns the status of the I/O point.

LDB(P,A,B) LDB addresses output point A of controller P and sets its state on or off depending on B.

RDW(P,A) RDW reads I/O word A of controller P and returns the integer value read.

LDW(P,A,B) LDW addresses output word A of controller P. The integer value of W is loaded into output address A.

CLR(P) CLR addresses controller P and clears all its physical output points to off.

EEM(P) EEM causes controller P to halt execution and Enter External Mode.

con't

LEM(P) LEM causes controller P to restart at location 0 and Leave External Mode.

ZER(P) ZER stores zeros in the memory of controller P.

LOD(P,N) LOD loads the memory of controller P with the content of file N.

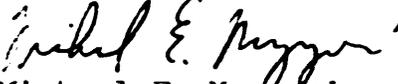
DUM(P,N) DUM dumps the memory of controller P into file N.

VER(P,N) VER compares the memory of controller P to the content of file N.

RDT(P) RDT checks the status of controller P after a ZER, LOAD, DUM, or VER function causes an interrupt, or reads output of an INDUSTRIAL 14 after it has interrupted a PDP-8.

Because the space these extra functions consume, some file buffers were sacrificed. Thus, only two files can be open at one time. Also, only device handlers co-resident with SYS: can be used with files.

Yours truly,


Michael E. Mazzoni
President

----- EXTENDED CHARACTERS IN EDIT -----

THE ONLY REASON THAT EDIT REJECTS THE CONTROL CHARACTER "N" IS THAT THE CODE 214 OCCURS IN A TEST LIST ON PAGE 0 IF LOCATION 35 IS CHANGED TO 7701 (THE VALUE AT 35 WHICH ENDS TESTING). THE CHARACTER "N" BECOMES NON-ECHOING, AND IS RETAINED DURING IV AND TE COMMANDS. IT IS RECOGNISED DURING A CHARACTER SEARCH, AND CAN BE DELETED AS REQUIRED.

I. M. TEMPLETON
DIVISION OF PHYSICS
NATIONAL RESEARCH COUNCIL OF CANADA
OTTAWA, CANADA K1A 0R6

3 ITEMS FROM IAN TEMPLETON

1) DEC FLOPPIES IN A SYKES SYSTEM:

THE DEC FORMAT OF INTERLEAVED, PARTIALLY FILLED SECTORS IS TOO COMPLEX TO BE DEALT WITH BY AN OS/8 HANDLER WHICH OBEYS THE TIMING RESTRICTIONS OF THE NON-BUFFERED SYKES 7100/7200 SYSTEM. I HAVE THEREFORE WRITTEN TWO PROGRAMS, DTOS (DEC-TO-SYK) AND STOD (SYK-TO-DEC) WHICH MAKE COMPLETE COPIES FROM ONE DISK TO ANOTHER, TRANSLATING AS THEY GO. WHILE THESE TWO PROGRAMS ARE WRITTEN TO USE A DUAL (7200) SYSTEM, THE COPY CAN BE MADE ON ANY FILE-STRUCTURED DEVICE (E. G. DECTAPE) FOR WHICH A SUITABLE HANDLER CAN BE INCORPORATED. I INTEND TO SUBMIT THIS PACKAGE TO DECUS, BUT ANYONE WHO WOULD PREFER THE PROGRAMS ON A SYKES FORMAT FLOPPY SHOULD CONTACT ME AT THE ADDRESS BELOW OR CALL ME AT (613) 992-2113.

2) NEW SYKES OS/8 PACKING HANDLER:

I HAVE MODIFIED OUR 16-BIT-PACKING SYSTEM HANDLER (DECUS 8-846) TO INCLUDE AN ENTRY POINT FOR UNIT 2, AND HAVE ALSO FIXED THE BOOTSTRAP TO RETAIN THE DATE. ANYONE INTERESTED SHOULD CONTACT ME AS ABOVE.

3) BATCH /T OPTION CHANGED TO /L:

THE /T OPTION IN BATCH PRINTS THE BATCH LOG ON TTY BUT TESTS FOR LPT ANYWAY. THE FOLLOWING PATCH REVERSES THE OPTION TO /L (PRINT LOG ON LPT) AND ONLY TESTS FOR LPT IF /L IS USED. THE LINK MUST BE CLEARED IF NO LPT IS FOUND AFTER /L, OTHERWISE THE LOG APPEARS ON TTY IN LPT FORMAT!

```
. GET SYS BATCH
. OD
307/4755  6211  CDF 10           /DF=1, AC=LK=0
310/5320  1765  TAD I OPW1        /1ST OPTION WORD
311/6211  6201  CDF 0           /DF=0, /L BIT IN AC11
312/1575  7010  RAR                /SET LINK IF /L
313/6201  7620  CLA SNL          /SKIP IF /L
314/0354  5320  JMP .+4             /NO /L, AC=LK=0
315/7141  4755  JMS I LPT1        /LPT?
316/7630  1346  TAD M65           /NO, AC & LK TO BE ZERO
317/1353   1346  TAD K65           /YES
346/XXXX  7713  M65,      -65
^C
. SA SYS BATCH
```

(ORIGINAL CODE FROM BATCH V5D)

I. M. TEMPLETON
DIVISION OF PHYSICS
NATIONAL RESEARCH COUNCIL OF CANADA
OTTAWA, CANADA K1A 0K6