



DECUS 12 BIT SPECIAL INTEREST GROUP  
NEWSLETTER

November

Number 19

1976

-----  
Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator

12 Bit SIG

c/o DECUS

146 Main Street

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 DEADLINE

-----  
Deadline for ready-to-use material for the next Newsletter is December 31, 1976. Material requiring editing/re-typing must be in earlier. Ready-to-use material should be prepared on 8 1/2 by 11 inch white bond paper. A one inch margin should be maintained on both sides, on the top and on the bottom. Material should be reasonably clean, legible, sufficiently dark copy for printing. Materials prepared on electrostatic printers (e.g. Versatec printers, Xerox machines, etc.) is often unsuitable for photographic reproduction.

SIG BUSINESS

-----  
The DECUS/US Executive Board has just informed me that they have voted to refuse our request for representation on the board. I think that a full discussion of the subject should be deferred until the next newsletter to give the steering committee a chance to talk to the board members during the Fall Symposium.

OS/8 DATE

-----  
The question of what is going to happen about the OS/8 date came up again a few days ago when Dave Kristol called. As everyone who has looked at the insides of OS/8 knows the date word format can only accommodate dates through the end of next year. The

monitor will not accept dates beyond 31-dec-77. Even if it would there is no way to represent the date in the 12 bits which have been allocated. With just 13 months to go till the date blows up it seems as though DEC will have to make a decision almost immediately on what to do and how to set it out to all the users of the system. Either some collection of patches will have to be published or a new release of the system will have to be ready and out to users by the end of next year. I have my doubts about the second happening in time considering the turn around time and cost for a new release. On the other hand patches may have to be fairly extensive and there may be a temptation to decide on a solution more on the basis of how easy it is to patch into the system than on how viable it will be in the long run. I understand that even in the big-buck world of DECSYSTEM-10 there was recently a similar crisis that was not fixed till after their date word ran out.

It might be that user input on how to solve the problem could be useful. Considering the problem of maintaining compatibility with the existing files as well as a large amount of user written software the only solution short of total overhaul of the date scheme and all the files and programs that use it is some sort of a compromise.

I would like for everyone who is interested to think about the problems and possible solutions and to send me any good ideas. To get you started consider the following approach:

1) To avoid the need to modify the way PIP, FOTP and many other programs treat the extra information words in directories not to mention many other difficulties I will try to stay with one 12 bit date word as far as device directories goes. This will maximize compatibility with existing files and file manipulation software like PIP and FOTP.

2) The current rule for creating an OS/8 date word is to subtract 1970 from the year. The result is required to be between zero and seven. This is why the monitor says 1/1/78 is a bad date. If we change the rule to say subtract 1970 from the year then take the result MOD 8 (i.e. mask it to three bits) then the resulting value will fit in the available three bits for the year. The years between 1970 and 1977 will remain unchanged. 3) Now we have to deal with how to distinguish 1970 from 1978. This is obviously the basic problem. The current rule for interpreting the date word says to consider the year bits as a value to be added to 1970. If somewhere in the system a longer version of the year were available we could change the rule to say that the three year bits defined dates in the eight years preceding today (or this year). For example a few bits could be allocated in location 07777 to define the bits just to the left of the current three year bits. They would be picked up by programs that knew about them, shifted into bits 7 and 8 and then added to the current three bits which go in bits 9 thru 11. The entire result would then be added to 1970. With two extra bits the year would be extended to 2001. By comparing the unsigned value of the standard 12 bit system date with the particular date word you are interpreting you could determine if

the date will come out in the future. If so you subtract eight from the result to make it a date in the past. All this sounds rather complex I am sure but it really is not and it has the advantage that with only a very small patch to the DATE function the system will accept the new dates. Programs that do not know about the new convention will still work but they will print out a date that is eight years back from the correct one for dates after 1/1/78. The advantage of this scheme is that unless you are playing games with dates in the future you always can represent the maximum range of eight years back from the present correctly. Other schemes that have been kicked around involve periodically redefining the base year (i.e. 1970) to be incremented by say four years. This approach sacrifices some of the range back from the present; it requires either the same allocation of bits in the system or patches to every program every time the base year changes and it requires the same comparison with the current system date to decide how to interpret the date in question.

If DEC would agree to some such scheme as this we could start modifying our programs now to be ready and patches to DEC software would be practical in the time frame available to at least assure a minimal solution before 1/1/78.

#### NOTE FROM CHASE AMBLER

-----

Chase Ambler has written about the charges for updates from OS/8 V3 to OS/8 V3C. He was charged \$400 for the update! I have checked with my local DEC office and found the following: 1) the price for an initial order of OS/8 has gone up from \$300 to \$400. 2) The update charge for licensed users is now \$150 - however you must use a special variation of the order number (it has an H in it) to get that price. Failure of Chase's salesman to establish this is the most likely reason for his trouble. Note also that according to my salesman the source update policy is still not set. According to him the present situation is that updates of sources are still charged for at full price! Any of you who agree with me that this is very unfair and unreasonable should make the loudest possible complaints to DEC. Chase is also looking for information on typewriter quality terminals. Any help you can offer would be of interest to many of our members so send a copy to me. Check the attached letter for details and address.

#### NOTE FROM HAL SINGER

-----

Hal (who puts out a very information filled newsletter about microcomputer topics with his students) wrote about a couple of things he and some of his students have been doing. First is a program called FILSAV - An OS/8 ASCII File Saving Program by David Bryant. FILSAV is a program to locate and retrieve ASCII material from devices where other programs (PIF or FOTF) are unuseable. Examples of use would be to retrieve files lost through deletion, directory failure or files with imbedded I-O errors.

With FILSAV the user may search entire devices for specified lines or phrases, perform directory lookups (when the directory exists), examine specified blocks and copy desired blocks onto another device. All these can be performed with a few simple commands making the retrieval of lost files an easy task. Hal says this program was motivated by having directories wear out on the floppy disks on the school's three CLASSICs. It has some of the flavor of FUTIL and some of the same function as the special patch to TECO for doing whole device searches for ASCII text but this program seems to wrap up the package for it's intended use very nicely. This is the sort of utility program you suddenly find your self in need of some Friday night at 9:30 when you start setting errors on the disc or tape with the only copy of something vital. When Hal submits it to DECUS it would be a good thing for many users to order so they will have it when the need presents itself.

Hal also wrote about some work Dave has been doing on EDU25. He added a floppy disc driver when DEC indicated it was not going to be able to do so. The addition yields a 4 to 5 user CLASSIC (programed in BASIC) which Hal thinks is "a fantastic system". If he could get the loan of 16K of memory and a terminal interface board from DEC he would make the mods that would result in an 8 user system with floppy BASIC file storage that could be sold for under \$20,000. Hal has given information on the system to his local salesman who will present it to DEC Educational Products. He hopes EDU will decide if it should go into DECUS or be supported in some other way. Perhaps the PDP-8 product line (which is still not really in the EDU part of the market) should give this a look in terms of it's broader markets too. Hal's address is Cabrillo Computer Center, Cabrillo High School, 4350 Constellation Road, Lompoc, California 93436.

NOTE FROM H. DOMINIC J. COVVEY

-----  
 Mr. Covvy has sent me a copy of the writeup for his SUPERFOCAL. He describes his program as follows: SUPERFOCAL is a superset of DEC's FOCAL 1969. It contains features found in OMSI's PS/8 FOCAL as well as additional ones for text manipulation. It has an integral stand-alone operating system working from pre-formatted disk cartridges. Two equal priority users are supported in a timesharing mode. A user continues execution until his time slice is ended or until he would have to wait anyway such as during terminal I/O. Each user has a separate copy of FOCAL with links to common routines in the executive. The user's copy of FOCAL and all status tables are core resident; however, some command overlays are stored on disk and are called as needed. These overlays are three pages long and at present there are two; one containing LIBRARY commands and one containing functions. The disk READ and WRITE commands are always resident. Program size is 30 pages or 7,680 6-bit characters. In addition, each program may chain to a total of 40 other programs which are stored on disk.

Mr. Covey also sent information on three other documents:

- 1) CLIFFS - The CLinical Focal File System. The application of Superfocal to the database area in support of a cardiac pacemaker clinic.
- 2) HEMO - A complete reporting system for a cardiac catheterization lab.
- 3) LP - A light pen/television data display system applied to analysis of left ventricular function.

Judkins from the write up and the description of the applications this sounds a lot like like MUMPS. The main functional difference seems to be that there is no equivalent to MUMPS' exotic structure for storing and referencing data.

The address is Cardiovascular Unit Computer Room, Toronto General Hospital, Toronto 101, Ont., Canada.

#### NAVY FOCAL

Harold E. Cronin has sent along information on a 12K version of OMSI FOCAL which he has developed for a PDP8e installation, which he calls NAVY FOCAL. Among it's features he lists:

- 1) Plotter and character generator for XY8e
- 2) Tape reader for GOULD 6000 datalotter
- 3) Display function for two color VR20
- 4) Reader function for 1/2" DEC mastape - TU10
- 5) Reader for Metrodata DL 622 datalotter
- 6) Function to read the switch registers
- 7) A virtual memory system using the RK8e disk to allow up to 21,420 variables

Harold is at The Naval Weapons Center, China Lake, California 93555.

#### NOTE FROM JOHN YOUNGQUIST

John wrote to call our attention to an interesting sounding publication which he recently received. The paper is entitled "Conversion of DEC PDP-8 Fortran to Interruptable Operation". It adds rings buffered interrupt driven handlers for TTY, HSR/P, and supports device 4, (LPT; and CDR; not supported). Also RT Clock and some special DF-32 support is included. The paper is over 60 pages of description, flow charts, and complete listings. It gets around the problem of auto-loading the interrupt stuff at locations 1,2 and 3 very nicely he says. To get this paper request:

AECL Publication No. 4663  
from:

Scientific Document Distribution Office  
Atomic Energy of Canada Ltd.  
Chalk River Ontario  
Canada K0J 1J0

and include \$3.50 in Canadian funds.

John has recently adapted a KV81 graphics system for his PDP8e. He would like to find someone who has a copy of EDGRIN and/or the documentation. He says that he can not get any information on EDGRIN from DEC. John's address is Verus Instruments Inc., Box

122, Fort Erie, Ontario, Canada.

NOTE FROM PHIL SIEMANS

-----

Phil sent along a copy of his Mode A EAE routines for FORTRAN II and he says that he has submitted them to DECUS. For those of you interested in the 23 bit format he uses to speed up the routines and how it fits into the design of FORTRAN II the write up documents the details.

NOTE FROM FRIEDEMANN BRAUER

-----

Friedemann wrote to say that he had revised Ian Templeton's "packed" format system device handler for the Sykes 7100 floppy disk. He has made it handle a dual drive, buffered system (Sykes 7250). He is planning to submit the system device handler FLOPSY (with entry points for both drives) and also a two page non-system device handler for both units called FLOPNS to DECUS. His address is HNO Forschungsstrakt, Klinikum Westend, Spandauer Damm 130, D-1000 Berlin 19, Germany.

MATERIAL FROM VERNON BLACKMORE

-----

Vernon has sent along the DECUS submission forms and write ups for a set of OS/8 programs he has developed. The submission tapes are to follow. They are quite interesting and I suspect that a number of you will want to look at one or the other of his submissions. First is KBLD which is a building program that converts OS/8 to use a KV8 display for the keyboard monitor, command decoder, teletype handler and ODT. Most OS/8 programs may easily be modified to use the KV8 for output. Patches are included for FORTRAN II (plus graphics routines), FORTRAN IV and EDIT. The EDIT patch also implements additional commands such as append from the high speed reader and line number listing. The other submission is called LOG. It has two main functions: 1) to record details on the use of the machine and 2) to allocate to each user a part of a mass storage device. Large storage devices may be split up into smaller separate devices which removes directory overflow problems and gives users their own individual disk space. The LOG system comprises three programs: LOG.PA - the source, MULT.PA - a special handler, and LOGANL.FT - an analysis program for the log file created by LOG. Also included are a source of his modified version of CCL which includes a number of useful commands and a copy of a beginners guide to his OS/8 system that might be of interest to anyone trying to write a similar guide for his particular installation. Until the tapes arrive and the submission is processed by DECUS you can inquire of Vernon at his new address for correspondence which is 18 Brookfield Road, Heaton Chapel, Stockport, England.

INPUTS FROM CHARLES LASNER

-----

I recently received a thick stack of material from Mr. Lasner. First, a summary of a write up on the PFS/8 monitor:

The Polytechnic Question Society Monitor System (P?S/8) is a complete program development system for the PDP-8 computer. Included are facilities for editing, assembling, and loading assembly language programs; and an extensive FOCAL programming facility.

The P?S monitor requires a 4K PDP-8 (or equivalent) CPU, a console I/O device, and a system device of at least 32K words. System device handlers are available for almost every PDP-8 mass storage device, ranging from devices as archaic as the LINC-8 LINCtape and DF32 disk to those as recent as the DSD/DYNEX cartridge disk system.

A major design criterion of P?S has been to provide the best possible level of performance for the user with the minimal hardware configuration. For example, in a benchmark test, a P?S assembly on DECTape took only twice as long as the equivalent OS/8 assembly on RK8 disk; whereas an OS/8 DECTape assembly took eight to ten times as long.

The system is comprised of: 1) The file system, 2) The keyboard monitor (and integrated system editor), 3) The assembler (PAL), 4) The binary loader (BIN), 5) The octal debugging technique (ODT), 6) P?S FOCAL, 7) The system utility programs.

Also included in the package was an offer of information about a pair of handlers for a high speed inter-processor buffer option designed by David Soersel of Syracuse Univ. One of the handlers is a system device handler which means that the remote machine can provide the actual system device. Speed is so fast (18 microseconds per 12 bit word) you don't really notice the effect of the remote SYS:. In fact he has connected a PDP-8e to a LINC-8 so that the LINCtape looks like it is connected directly to the 8. Another configuration using the IPB is to run LAF6W (on the LINC-8) on a VT-8e (on the 8e) by patching PROGOFOP to do I/O to the IPB rather than to the TTY. Also included is much more on good things being done on the LINC-8 like 4000 block (128 words/block) tapes that work on PDP-12s. Write him for more info.

Another item is two DECTape handlers. TC08SY.PA is a system device handler for TC01/TC08 DECTape that provides for both DTA0: (SYS:) and DTA1:. The other handler (DTX) is a two pass handler for TC01/TC08 that provides for 8 file structured (single tape) entry points and also 8 non-file structured (multitape) entry points. The non-file structured operation unloads a tape after it is full and waits for the next tape to be mounted then continues. This last would be nice where you want to output something like a listing to tape but it is too long for a single tape.

Also included is a long list of complaints about the RX01 floppy disc system with a claim that there are serious error conditions which are not handled properly and which he says cannot be handled (for OS/8) due to hardware limitations and limited space in device handlers.

The address for further information is P?S N.Y., 72-55 Metropolitan Ave., Middle Village N.Y. N.Y., 11379.

NOTE FROM CHRISTOPHER F.F. MACKAY

Christopher writes to say that he has worked out a bootstrap that goes on block 0 of a DECTape so that when the TD8E ROM bootstrap is started, his RX01 is booted. This is a neat way of saving the cost of a new bootstrap ROM when you upgrade your system. He is interested in hearing from anyone who has had experience in linking a Nuclear Data ND100 Multichannel Analyser to an Omnibus 8. His address is The Royal Sussex County Hospital, Brighton, BN2 5BE, U.K.

NOTE FROM HANS PETER STOEHL

Hans wrote with a couple of questions: Has anyone tried to handle both upper and lower case characters in OS/8 FORTRAN IV? He says it is easy in FORTRAN II. He would appreciate any help or he will collaborate with someone. He is also interested in hearing from anyone who has a DOUBLE PRECISION software package for either OS/8 FORTRAN IV or FORTRAN II. On both these questions I would like to get a copy of any information you have because these questions come up frequently. Hans' address is c/o Institut für Modellstatik, Universität Stuttgart, 7 Stuttgart 1, Keplerstraße 11, Postfach 560, Stuttgart, GERMANY.

NOTE FROM BRIAN CONVERSE

Brian writes:

"Two years ago when I first started to use a PDP-12, I found that the console was a little bit weird. As time went on, and I became more familiar with it, I wrote off the console as an economy measure or as an exalted concession to LINC-8 and classic LINC users. I began to notice other things as I began to use the disk, mastape and FPP - there weren't any lights! The FPP is buried down in there and it just laughs at you.

Now the classic LINC had a few features that still haunt me. For one thing, there was a light for everything in the "standard" configuration. None of the manufacturers (US Gov't, Spear, DEC, and a few university operations) offered other than the "standard" configuration. So everybody had all the basics; none of this like our TU-68C (PEC 6X60) 9-track drive, serial number 25 that I have to xerox the manual for to explain differences with the TC-58 or to prove DEC made them. And those console lamps looked sexy, relamped immediately, and lasted for years (28 volt lamps running on 15 volts). The 23 classic LINC's the government made had teflon (c. 1963) insulated wire in the console unit and really good switches. It's hard to imagine Dick Clayton willingly allowed the PDP-12 to have such a horrible console. It's hard to imagine what anguish and compromise Wesley Clark had to go thru to come up with a salable LINC-8.

And it's hard to believe Dan Smith can E-stop at 40 and



exam/deposit and live. My experience has been that E-stop and F-stop STOP right in the middle of an instruction and any exam/deposit (with or without stepping) can foul things up. Watching the lights, I think what happens is that there is a mem addr to PC transfer that is part of what classic LINC called GGNI sequence: "Go Get Next Instruction". Since you've changed the memory address, that address goes into the PC and the PDP-12 goes off into OZ somewhere. The reason is that the PDP-12 doesn't finish the instruction cycle - it just hangs there with an unfinished cycle like an idiot. Now I've done marvelous things because it doesn't, but I did wonderful things on a classic LINC, which did, and given the choice, I'd prefer the LINC's way. The logic density on the LINC was zero compared to the 12 (one LINC card equals about one IC in the 12!) yet LINC was smart enough to finish out the cycle when you hit EXAM! The 12 should've had a superior console, like E-stop on the 23rd reference to location 345 or when location 345 equals 0007. The instruction trap should have a (displayed on console) mask register, etc., etc. I hate DDT. In fact, I've never used it. The only place for software DDT is in microcoded machines where it's invisible to the object code.

I've got a problem - I'm using PDP-12 FAST SAMPLE mode with the API on servicing the KW12 clock. The "background" code is an 8-channel display of signals on 8 A-D channels being sampled every clock interrupt. Additional A-D channels are sampled upon schmitt trigger interrupt. The displayed channels can be positioned by the 8 analog knobs. The problem is, I get interrupts from the clock which cause sampling that messes up the knob values at times because of the asynchronous nature of the interrupts. I'd switch to regular sampling but I'm in as tight a loop as possible now and can't afford any more flicker. So the background "display" has this random "fuzz" all over it. Anybody got a way to read the current mux channel on the 12? (I'll have to do it in the relatively infrequent clock interrupts, and I rail at having to pick interrupted PC out of API stack and back up thru code looking for the last SAM - the only solution I can think of)

Also to Dan Smith - take a look at PDP-12 manual for EXAM/DEPOSIT ritual to beat all. Makes single-stepping with PDP-12 look like 15th century torture. Now, in our case if you turn on API...!

One day I have always planned to write a program called "console on scope" for the 12 with a light pen. This would allow real fancy console design. In fact, one day I hope to see an LSI-11 type machine with microcoded or micro-computer driven CRT "console" display that you can rearrange (you could have a flip switch to go to regular display mode or even have superposition of hardware-generated "console" on software-generated displays)."

Brian's address is MO 58, Environmental Protection Agency, Research Triangle Park, North Carolina 27711.

THE ASHEVILLE SCHOOL  
 ASHEVILLE, NORTH CAROLINA  
 28806

Oct 8, 1976

Mr. Bob Hassinger  
 Liberty Mutual Research Center  
 71 Frankland Road  
 Hopkinton, MA 01748

Dear Bob:

First of all, may I express my personal appreciation to you for your fine efforts with the OS/8 newsletter? I know how much work it must involve in getting it out, and I must be one of hundreds that you know nothing about that read every word....and profit by it! Many thanks!

Thought you might be interested in the enclosed. The \$400 really frosted me, especially since we had VII. I had to raise the additional \$300 by teaching a BASIC class to the public on my own time and also contributing \$100 of my own money.. Then they bill us for \$400 after we had sent \$100.

I definitely will not go to any future version at these prices (if we keep Ver IIIc). I don't know what will happen, but us little guys can't afford updates this often. I have an 8E with 8K and dual TU56 etc.. My whole budget for the year is \$600, which includes ALL maintenance! DEC is rapidly becoming a four-letter word.

We are presently using the VT05 and a tty. The video is console device, and the TTY is used as LPT. (However, I can mechanically switch device codes and swap them around). Since the TTY has about 3,000 hours on it, I have been looking for a new terminal....and especially one that would be suitable for "personal" form letters from our administration. In other words, no dot matrix or special impact paper things.. The Selectric seemed to fit the bill, so with the help of a kind parent with \$1000, we ordered the Anderson-Jacobson 841 Selectric. They assured us all we had to do was "plug it in", as we had EIA interfaces. (but maybe change the baud rate, which I could do). It plugged in fine, but obviously to me after we got it, ~~think~~ it wouldn't work as it wasn't ASCII, or even BCD, but something called Correspondence Code, whatever that is. So, I packed it up and shipped back.

Do you know of any LPT or terminal (used at about 1 to 2,000 dollars) that would work?? It must be a "typewriter" face, and at least 30cps, and I guess, accept ASCII..... (I know very little about such things as codes, but am a whiz with BASIC!) Any help you could give us would be appreciated...

Best regards,

  
 Chase Ambler

## NOTE ON THE .DEASSIGN COMMAND

WITH THE ADVENT OF OS/8-VERSION 3 (AND NOW 3C) CAME THE DEMISE OF ONE OF THE NORMAL SYSTEM COMMANDS: THE .DEASSIGN COMMAND. IN PREVIOUS VERSIONS OF OS/8 THIS COMMAND COULD BE USED TO CLEAR ANY USER ASSIGNED DEVICE NAMES WHICH MIGHT INTERFERE WITH THE PROPER OPERATION OF THE PROGRAM. I FREQUENTLY USED IT TO CHANGE THE ASSIGNMENT OF DSK FROM ONE TAPE TO ANOTHER. HOWEVER, UNDER VERSION 3 THE .DE COMMAND IS ONLY AVAILABLE UNDER CCL (IN SPITE OF WHAT THE OS/8 HANDBOOK SEEMS TO IMPLY) AND WITH CCL DISABLED (OR EVEN REMOVED, PERISH THE THOUGHT - BUT IT SAVES A LOT OF SPACE!) IT IS POSSIBLE TO .ASSIGN DEVICE NAMES, BUT NOT TO .DE ASSIGN THEM. CLEARLY THIS IS QUITE ANNOYING, BUT EVEN WITH CCL ENABLED IT IS NO LONGER POSSIBLE TO EXIT FROM A PROGRAM, DO A .DE AND THEN RESTART BECAUSE THE 'NOT-RESTARTABLE' BIT GETS SET AND I SUPPOSE THAT THE PROGRAM ITSELF GETS CLOBBERED SOMEHOW. ALL THIS IS MOST UNSATISFACTORY, BUT FORTUNATELY THERE IS A TRICKY WAY AROUND IT: THE BREAK COMMAND IN ODT USES THE ASSIGNMENT TABLE FOR ITS OWN PURPOSES, SO YOU CAN HIT CTRL/C, TYPE .ODT, THEN 'B', ANOTHER CTRL/C, AND A .ST TO RESUME THE PROGRAM WITHOUT CLOBBERING ANYTHING. THIS IS ALSO A LOT FASTER THAN USING CCL TO DO A .DEASSIGN - AT LEAST ON A TAPE SYSTEM!

I have lately been working with RALF code often generated via the FORTRAN IV compiler and I found that in these cases the tools provided by DEC for the programmer were inadequate. I have therefore done the following to fix this.

- 1) The FORTRAN II compiler has a neat trick of inserting the FORTRAN statement into the RALF sources as a comment. I wrote a small FORTRAN program which will do this: It takes a FORTRAN listing file together with a RALF source file (can be produced in one compilation by the syntax .Compile TEST.FT/A-LS). The program will also delete the statement numbering produced by the FORTRAN compiler if you wish. The statement numbers have to be included in the compilation as they are used by the FORT program to synchronize the FORTRAN and the RALF sources.
- 2) CREF contains an option to support RALF. The support for RALF is, however, very poor in the standard version. I decided to see if I could do anything with a reasonable amount of work. I am not going to recode CREF. I wrote a patch which will do the following:
  - a) It adds several but not all FPP symbols to RALF. This means that crossreferencing a FORTRAN output does not include several hundred references to FADD, FLOAD.
  - b) I made  $\#$  a legal character but it is treated by CREF and output in the CREF table as  $[$ . This is not a major difficulty, however. Several problems remain when using CREF with RALF code but I think these problems are, if not acceptable at least such that it is still worthwhile using CREF. I think DEC could do a bit of work on CREF to fix it.

/FUNCTIONS:

```
/1 ADDS SEVERAL BUT NOT ALL FPP SYMBOLS TO RALF
/   THE SYMBOL #LIT IS ALSO ADDED TO REMOVE MANY
/   (IN LARGE PROGRAMS SEVERAL 100) REFERENCES
/2   # IS A LEGAL CHARACTER BUT MAPPED(AND OUTPUT) AS [
/
/U /M WORK IN COMBINATION WITH /R (/P=/U)
```

/PROBLEMS REMAINING \*\*\* DEC DO SOMETHING\*\*\*\*

```
/A   RALF ALLOWS TAGS TO HAVE THE SAME NAME AS AN OPERATOR
/   THE CODE JMP JMP IS POSSIBLE IN RALF
/   IF A FORTRAN VARIABLE IS GIVEN SAME NAME AS A PERMANENT
/   SYMBOL (EG JA OR KSR)IT IS NOT REFERENCED
/
/B   RALF ALLOWS A SYMBOL TO BE FOLLOWED BY A , IN
/   POSITIONS WHEN IT IS NOT A DEFINITION
/   EG JA #100, 5
/   THE LAST SUCH OCCURANCE WILL BE FLAGGED AS
/   DEFINITION
/C   CREF DOES NOT RECOGNIZE THE RALF SYMBOLTABLE
/   ALL NAMES IN THE TABLE(INCLUDING SECTION FLAGS AS X)
/   ARE CROSSINDEXED
/D   SEVERAL FPP SYBOLS AND MOST RALF PSEUDOOPERATORS ARE NOT
/   DEFINED THIS DOES NOT INFLUENCE THE FIV
/   PROGRAMMER BUT WELL THE RALF CODE PROGRAMMER
/STILL IT IS BETTER THAN WITHOUT THE PATCH
```

The comment transfer program (COMRAL.FT) and the overlay to CREF can both be obtained from me. They will be submitted to DECUS later when I make a new submittal of my bits and pieces tape. Note that in the CREF overlay the DEC patches are included. If you want to do any patching besides that done by the overlay, you have to be careful as there is very little room left in CREF.

Further notes re FRTS and BASIC with a non-TD8E 2-page system handler

The latest versions of FRTS (V4) and BASIC (V4A) operate slightly differently from those (FRTS with no version #, BASIC V 3.0) described in Newsletter #16. The following modifications should be made to those notes.

1. FRTS (V4)

Here the TD8E handler is recognised by finding 0003 in the field 0 address stored in 12756. Thus 12756 contains 7612 and 12755 contains 7775. The instruction-changing operations start one location earlier, i.e. the TAD (CDF MAX) occurs at 12674, but the subroutine itself is unchanged.

2. BASIC (V4A)

Again, the 0003 in 07612 of the TD8E handler is used, but otherwise operations are more or less as before except for some minor changes of address.

Thus 567, 2556 and 7374 of BLOAD.SV contain 7570, to test the system device code; 566 and 2555 contain the address 7612, and 565 and 2554 contain 7775. The test for 6223 in 7642 is still there at 7372, and the special addresses for changing (& testing) are in 557-61 and 7370-73. The change routines now start at 503 and (as before) 7256. The operations in BASIC.FF are unchanged.

I. M. Templeton  
National Research Council of Canada  
Ottawa, K1A 0R6, Canada

Notes from Ernst Lopes Cardozo, Utrecht, Holland.

## CHAIN & RETURN

One of the things I always missed in OS/8 is the possibility to call a program as a subroutine, i.e. a chain with return. It turns out that this feature can easily be implemented.

As you may have noticed, the last location of field 1 is not used by OS/8. After bootstrapping the system this word contains a zero. To call a program you should LOOKUP the savefile of your own program (which must be on SYS:) and deposit the starting blocknumber in location 17777. Next you LOOKUP the program that should be called and perform a normal CHAIN to it. When this program is finished and returns to the keyboard monitor (7600 or 7605) OS/8 will chain back to your program. To have this feature, the following patch must be applied to OS/8 (the format of SPY is like ODT, with blocknumber.offset):

.R SPY  
\*SYS:

```

11.1/ 6211 5277(CR)          JMP PATCH
11.77/ XXXX 6211(LF)        PATCH.  CDF 10
11.100/ XXXX 1663(LF)       TAD I (7777
11.101/ XXXX 7450(LF)       SNA
11.102/ XXXX 5202(LF)       JMP back to normal
11.103/ XXXX 3311(LF)       DCA BLOCK
11.104/ XXXX 3663(LF)       DCA I (7777
11.105/ XXXX 6201(LF)       CDF 0
11.106/ XXXX 6212(LF)       CIF 10
11.107/ XXXX 4422(LF)       JMS I USR
11.110/ XXXX 0006(LF)       CHAIN
                                BLOCK.  0

```

Note that after return to your program location 17777 will be zero again. Application if this mechanism revealed that BASIC does modify the top four locations in field 1 (both the old (V3) and the new (V3C) versions. As far as we could unscramble the code, BASIC.FF tries to save and restore the top four locations in the highest field, which are used by BATCH. Unfortunately the save operation is (sometimes ?) inhibited so that the restore moves the first four locations of the command decoder area to 17774-17777. The following patch cures the problem:

```
.GET SYS BASIC.FF
.ODT

14352/ 1020 5742
^C
.SAVE SYS BASIC.FF
```

PAL8 V10 the symbol table.

This is a comment on the article of J. van Zee in the newsletter #18. As documented in the OS/8 System Release Notes (V03C) the new version does interpret the /K switch, but in a different way. Without /K the program will keep the USR in core to speed up assembly of small programs. In that case you have room for 636 symbols. With /K you get 897 symbols (all in 8K).

#### STANDARDIZATION of RUBOUT SEQUENCE

Now that OS/8 is frequently runned on virtual machines, i.e. in the background of a realtime and/or timesharing system, the need for standardization of the RUBOUT mechanism is stronger than ever before. Especially in systems where the OS/8 job is swapped between memory and secondary storage, echoing of keyboard characters must be done by system software in order to avoid unacceptable delays. In general this is not difficult and its is very well possible to detect which output characters coming from the background program are echo for the input characters. So the system software may drop the superfluous echo characters. However, in case the input character is RUBOUT, nothing can be said about the reaction of the background program. A quick inventarisation reveals the following:

OS/8 (Keyboard Monitor and Command Decoder): echo backslash if this is the first RUBOUT in sequence; Echo the deleted character; Echo backslash before the echo of the first normal charater after a series of RUBOUT.

OS/8 BASIC: echo backarrow (underline) for each RUBOUT.

EDIT, BASIC runtime input, FORTRAN IV input: echo backslash for each RUBOUT.

TECO: echo the deleted character.

With video terminals rapidly filling the world, a number of new possibilities have emerged. Those terminals supporting the BACKSPACE function can be driven with backspace, space, backspace to remove the deleted character from the screen. It is one of the advantages of the virtual machines that the foreground system can give a reaction adapted to the type of

terminal attached, while the OS/8 system can be left unmodified. However, this is only possible if the standard reaction of OS/8 and other background programs is easily detectable for the foreground software. I think it would be wise to return to the old standard of a single backslash for each RUBOUT received.

#### A BUG IN RALF V60A (FORTRAN IV)

RALF has a problem with modules whose ESD requires more than one output block. Such modules will result, e.g. from a FORTRAN subprogram with references to 50 or more common blocks.

The problem is that incorrect relocatable will result if any of the RALF I/O devices use 2-page nonsystem handlers. (I've verified this only for the list output device, but think it's true for the others as well).

The problem can be avoided by having all files reside on SYS:.

The problem is due to a section of code (location 6400, NEW &c.) which determines whether there are any other handlers and, if so, runs around patching all locations that refer to buffer areas. Unfortunately, it misses one (the constant (-OUBUF-377 at location 5717 in ESDBUF). This problem will be reported to DEC and seems easily fixable/patchable.

To determine if the problem exists in your configuration, compile the program

```
COMMON /BUG1/A1
COMMON /BUG2/A2
.
.
COMMON /BUG50/A50
STOP
END .
```

request a listing with the LOAD, and see if the map includes all 50 blocks or not. In our system, which has a 2-page (special, local) line-printer handler, the problem will occur if compiled with a /F and not if not.

--Dan Smith  
 Eye Research Institute  
 20 Staniford St.  
 Boston, Mass. 02114

Oct. 29th, 1976





DIGITAL EQUIPMENT COMPUTER USERS SOCIETY  
MAYNARD, MASSACHUSETTS 01754 / TEL. 897-5111 / TWX 710 347-0212

DECUS LIBRARY NOTE

DECUS has received several inquiries from users for the list of programs in the library on diskettes (RX01). The DECUS library presently has very few programs to offer to these users. We encourage users to submit programs on diskettes to help facilitate the requests being received.