PS 8
OS 8 Sig
OS 12
DECUS

October Number 14 1975

Contributions and correspondence should be sent to:

Bob Hassinger, Coordinator
OS/8 Special Interest Group
c/o DECUS
146 Main Street
Maynard, Massachusetts 01754

MACREL

The latest information on the progress on the MACREL project is a bit discouraging. Apparently in the last couple of months a decision was made to stop work on MACREL and to shift resources from that project to an effort to get caught up on the SPR backlog which seems to have grown to be quite substantial. In addition, apparently, the plans for the OS/8 version 4 release which was intended to include MACREL have been delayed. In place of the version 4 release the plans now call for a version 3Brelease which will apparently contain nothing but maintenance updates that result from the SPR work. If you have any bugs in any of the OS/8 software you should get your report in immediately so that it can be included in the version 3B release.

PDP-8/12 SOFTWARE MAINTENANCE

As noted above the single largest problem users have been having in recent months has been with bugs in DEC's software and slow/inadequate SPR service. (Example: I received an answer to one of my 29 April SPR's on 29 September - 5 months to the day. All it said was that I was correct and that they would publish the way to get around the problem that I had included in the SPR. They could not even type the example correctly. Most important they made no effort or promise regarding making the problem go away or to keep it from crashing programs at run time!) I have scheduled a session at the Fall DECUS Symposium in Los Angeles to talk about this subject. DEC representatives from the responsible areas have committed to participate in the session. They will not be able to say what has happened to SPR number so-and-so but they will be addressing how DEC supports our software, what the problems with that support are and what can be done to improve the situation. Bring your problems and ideas. I intend that the session be an interactive, open forum.

MULTI-8

I recently received a copy of the first volume of detail information on the MULTI-8 system from J. F. Anthoni and E. Lopes Cardozo. The report seems to be

well written and it describes a very interesting system combining a real-time foreground with a multi-user time sharing background. The system is still under development. It presently exists in versions to run on a standard PDP-8, an 8I with the intelligent instruction trap hardware that has been described in European DECUS Proceedings by Anthoni, and it also now works with a PDP-8e intelligent trap built by Mr. Cardozo at Utrecht. This is similar to the 8I trap except that it is programmable. Mr. Cardozo also indicates that he is proposing a more sophisticated "memory management unit" that would improve the performance in user mode. All processor generated fields would be translated by a RAM table. Thus, the background program would run in truly virtual memory and a paging strategy implemented on a per field basis could be implemented. Also this unit would involve a programmable untrap so that individual CDF instructions could be enabled or disabled at run time. Some of this hardware is a little like what EDUCOMP has implemented, I think.

A note just arrived from Ernst Lopes Cardozo with some further information on MULTI-8. He says the system will be distributed by a small Dutch software house. They were pushing to have a distributable system at the DECUS EUROPE Symposium on 10-12 September. A special workshop was planned. No further information yet on cost or who the software house is.

They use an interesting scheme to reduce trapping overhead in this system. When a skip IOT is trapped (i.e. TSF, etc.) it is patched to be a "SKP" to eliminate future traps. (If the monitor can't handle further I/O to the device then it will temporarily suspend the user when he tries to do the further transfers (i.e., TPC, TLS, etc.) anyway so this patch which makes it seem that the device is always ready does no harm). This handles the problem of a program hanging in an I/O wait loop undetected, and it reduces the number of traps that need to be serviced by a factor of up to two in many cases.

ETOS

EDUCOMP's multi-user OS/8 system called ETOS seems to be coming along fairly well now. I recently had a chance to review their System User Guide and System Manager Guide and it seemed to have a number of useful features and seemed to be reaching a level of some maturity. I hope to get a chance to go down to their site soon and give the system a try. If so, I will report on how it works.

NOTE FROM PAUL DIEGENBACH

Paul has written in response to the suggestion of exchanging program material that is not yet at a point where it is ready for a formal DECUS submission. He thinks that he could help in this sort of thing in Europe. He volunteers to make informal copies of DECtapes or LINCtapes. You can contact Dr. Diegenbach at University of Amsterdam, Zoological Lab., Plantagedoklaan 44, Amsterdam, Netherlands.

Dr. Diegenbach also suggest that a way to deal with letting users know about updates to library programs after they have ordered them would be to publish a list twice a year which shows what programs have been updated. He thinks that

this would be adequate for letting people know about new versions of programs they are using. He suggests that then all a user would do would be simply to order the new version. This is as opposed to some sort of more sophisticated automatic system. What do you think of this idea? Would it be adequate to meet the need for informing current users of updated DECUS software?

GERMAN LANGUAGE FOCAL HANDBOOK

Rudi Stange from Germany has sent along a little note describing the latest issue of a German language FOCAL programming handbook which he wrote back in 1969 and has re-issued a 1970 and 1973. It's a very nice book. I wish we could find a way to have it translated and published in English. It appears to be the most comprehensive book on the subject that exists.

Rudi's description of the book follows:

German language FOCAL handbook (3rd edition)

It lists among other things differences between FOCAL-68, 69, FOCAL-W, OMSI-FOCAL, PDP-11 FOCAL and FOCAL-GT. It shows symbol tables and many actual examples (copies of console printouts).

It offers an exhaustive introduction to FNEW incl. internal handling. It explains multiuser FOCAL (QUAD, LIBRA) and CLINE. 230 pages, price DM 10.--plus handling.

If there are not too many requests, I will be glad to send one copy free of charge to each inquiry within Europe.

Rudi Stange
c/o DIGITAL EQUIPMENT GMBH
Abt. Sales Support
D 8000 Munchen 40
Wallensteinplatz 2

NEW PROGRAMS IN THE DECUS LIBRARY

12-193 A set of FORTRAN/SABR routines for the DF32 - Rudolf Albrecht and Helmut Jenkner - University Observatory - Vienna

8-769 SELF DRILL - Advanced 8K version of 8-656. Set of general purpose learning algorithums. Prof. Wheeler - Beloit College, Beloit, Wisc.

8-772 OS/8 compatible VC8e handler for mass storage systems. Steven V. Bechtolsheim - Max Plank Inst.

8-773 Graphics Package for the Tektronix 4010 terminal under OS/8 - James Lerner - Union College, Schenectady, N.Y.

8-778 PFCF Polynomial function curve fitting FORTRAN II. Pei-Nam Tsung - Buffalo General Hospital.

8-779 TC58.PA - OS/8 version 3 device handler for the TC58 Magtape. Peter Lempkin - NIH.

8-780 SPLIT & SPLICE - Pair of programs to split up large files into several smaller ones and to recombine them. Useful for such things as taking a large listing on an RK disk and braking it up into parts that will fit on DECtapes, cassettes, or floppy disks for storage, backup, or transfer. Rev. Chase - Portsmouth Abbey School.

8-782 DEVHND - Device handler of a storage scope interfaced through the AXØ8. Robert V. Kenyon Jr. - University of California at Berkeley.

8-783 EDITV - OS/8 EDIT (V3) with View on an AXØ8 interfaced storage scope. R. Kenyon (see 8-782 above).

RANDOM ACCESS I/O FOR FORTRAN/SABR

John Algeo has sent along an abstract for a package of OS/8 FORTRAN II routines that he has written. He feels that these are in the "not well enough documented for DECUS" category although reading his abstract and documentation I would say that his material seems to be as well documented as most of what you will find in DECUS. His abstract is as follows:

> A group of subroutines have been developed which implement random access file input/output for programs written in FORTRAN II or SABR. Record lengths from 1 to 256 words are allowed, and all data is handled in core-image format. The system is currently configured to allow five files to be active at one time; however, this number may be expanded via some trivial modifications. A feature of the system is its ability to keep files open across calls to CHAIN. Routines are provided to CREATE, OPEN, READ and WRITE files on any directory device.
>
> The package was designed to be reasonably efficient, and, although it has not been timed against a FORTRAN IV program, I believe that it should be quicker than the IV-level direct access routines.
>
> An indexed file handling package based on these routines is under development.
>
> Documentation and source listings are available from the author.

Unfortunately, I do not have an address with this note, however pending catching up with John's address you can at least contact me for a copy of his more detailed writeup, if you are interested. As soon as I find his address I will put it in the next Newsletter.

NOTE FROM REV. CHASE

Rev. Chase says he has been working along the same lines as Bill Kaufman on a PDP 8/E EAE version of multiply and divide for FORTRAN II. His is full triple-precision. He is just to the initial testing phase.

He sends along an EAE random number generator for FORTRAN II. It is too long to fit in the Newsletter this time but if you want a copy let me know.

Rev. Chase's source DECtape with some of his offerings to the OS/8 world has sources for undocumented programs in addition to the DECUS published material. For example, REWDIR which he uses to rewrite directories and to create system-type empty directories on a TD8e drive not currently acting as SYS:

He reports having trouble looking at and changing locations 200-377 of a two-page handler with BUILD (with patch up through the SIZE modification).

LABORATORY BASIC

Stanley R. Vivian sent a copy of his very nice manual for his modified version of OS/8 BASIC. His software is device independent and is set up for several PDP-8 and PDP-12 configurations. He says there is a nominal charge for the software - payable to the University of Manitoba. If you are interested I suggest orderingthe User's Manual for $5.00. See attached writeup for details.

RUN TIME FILE MANIPULATIONS FOR FORTRAN IV

I just heard from Bob Phelps that he has succeeded in writing a FORTRAN IV callable subroutine named "USR" that can do all the things necessary to open and close files while running programs. This is functionally similar to the IOPEN,""OOPEN" and OCLOSE" routines for FORTRAN II.

Bob has promised to send me a copy of his program. When I get it I will report further.

NOTES FROM BILL HAYGOOD, JR.

Bill has sent along a couple of notes recently.. First, he is looking for someone who might be interested in selling either a TD8e or TC08 DECtape. He owns a system himself and would like to add the DECtape capability to it. He is presently working on the design of a multi-user system for OS/8. In that connection he has also sent along an article on his thoughts regarding the question in the last Newsletter about standardization of IOT codes time sharing and foreground/background systems. This will be attached to the Newsletter.

INQUIRY FROM PUSTY WHITNEY AT OMSI

While I was talking to Rusty recently he indicated that OMSI was interested in PASCAL for the PDP-8. The OMSI people would be interested and willing to work with anyone else who was interested in this project.

NOTE FROM WALTER C. DAUGHERITY

Walter reported an SPR that he submitted the 28th of July regarding OS/8 BASIC. He says that if you attempt to compile programs containing constants that exceed 10 or 11 significant digits (depending upon the particular constant) the number is compiled completely incorrectly. If you write a program that says:

```
A = 0.5000000000000
B = 0.3333333333333
PRINT A,B
```

you will demonstrate this. The values printed are quite startling. Walter is interested in exploring the use of the TD8e DECtape for reading and writing LINCtapes. He suggests that if someone could work up a little program that he has outlined and use it to put a suitable pattern on a LINCtape that he could borrow for while to test his program, he might be able to get a LINCtape handler running on the TD8e. If you are interested in working with him you can contact him at ECRM, Inc., 205 Burlington Road, Bedford, MA 01730, telephone (617) 275-1760.

NOTE FROM BILL LENON

Bill is working with DECUS 8-747 (STAGE 2 & FLUB). He found a bug in the memory sizing routine.

The fix is:

Location	From	To
06412/	7326	7327

The source code change is in S2RTS.PA at location STARTUP + 12_8:

From:

```
CLL     CIA     CML     RTL
```

To:

```
CLL     CIA     CML     IAC     RTL
```

He is having trouble compiling FLUB on a 16K system and will have to use at least 20K. Otherwise he says it works OK.

LETTER FROM NORMAN R. DOTTI

Mr. Dotti writes to tell us of some problems that he's found with FORTRAN IV. I will attach a copy of the SPR itself. As space does not permit reproducing all of the pages of examples you can contact me if you need a copy of the details.

Mr. Dotti says that one of these problems took four months to solve. One of his problems concerns the CLOCK function. It did not work properly in version 1 but a patch to FRTS fixed things up. However, when version 2 was released the CLOCK function again did not work. This time it took a new clock module which was to be added to FORLIB using LIBRA. DEC told him that they do not intend to make this fix available to the general public so if someone really needs it they can send him a DECtape. He could make paper tapes with a teletype, however that's not very attractive so he tries to avoid it. With this fix the clock seems to work correctly he says. His address is National Loss Control Service Corporation, Long Grive, Illinois 60049.

His second problem involves the data statement in FORTRAN IV and the SPR for it is attached. He suggests that if you think you're having a problem like his that you should make some tests on the data statements to insure that they are working properly.

His final comment is as follows: "I have just written a letter to DEC through our Sales Engineer regarding the support of OS/8. DEC seems to be phasing out or at least down, their support of the PDP-8 in spite of the fact that a lot of us depend on it. I would really like to see them come up with a supported super OS/8 (SOS/8) like that discussed in the July Newsletter. I don't expect software to work right the first time (but it would be a pleasant experience) but I simply must have more realistic responses when problems do develop. My work depends on it and I cannot afford to have it effectively out of commission for months at a time while an SPR floats around."

Mr. Dotti says if anyone has comments on any of the above, he would appreciate hearing them.

NOTE FROM LARS PALMER

Lars has sent along a couple of items for the Newsletter. The first involves a bug that he has found in his program EXPIP:

A bad bug exists in EXPIP version 7. It concerns the operation of /M and /P options and on the RK8E disc. In this situation the output file size is sometimes calculated wrongly. The following patch should be implemented and corrects the situation.

Location	Old	New
12133	7041	7710
12134	7510	5337
12135	7041	7350
12136	7350	3330
12137	7001	2330
12140	3330	7000
15613	6701	6703

Old locations might not be these listed.

Lars points out that there is a special configuration that offers particular problems to RTS8 users and to multi-user systems. That is, the FPP-12 option. Tom McIntyre has addressed this problem to some degree in connection with his PDP-12 at the last one or two DECUS Symposiums. It is a special problem because the normal schemes for relocating the fields for background programs don't work in the case of the FPP-12. Any configuration that has this hardware would very much like to be able to use it in whatever system they implement.

Lars sends along the following SPR comments. These are SPR's that he has submitted and so far are unanswered. Regarding the RALF SPR mentioned in the last Newsletter, my explanation was partly correct but not fully true. (A) He has a rather complex FORTRAN IV program that runs beautifully under FRTS version 2 but constantly gives input error under version 3. No explanation of this so far. (B) The patch number 5 to FORTRAN inhibits in at least some cases the function statement capacity. (C) There are at least two cases in the compiler where the compiler refuses to behave properly on errors in the FORTRAN source code. The error:

```
DO 10 I = J-1
```

is not detected by the compiler. The error:

```
LOGICAL L
If (L) 10,20,30
```

produces halt in the compiler. At least his compiler halts. DEC says that they cannot reproduce this problem.

NOTE FROM JIM VAN ZEE

Jim wrote about an SPR he submitted. It involves the /I option of ABSLDR. It seems that it does not work correctly when the core image contains full fields. The patch to correct this should be in the October Digital Software News. Jim has noted that the same bug appears in all versions of the system all the way back to PS/8 and wonders if the fix should be verified or adapted for the older versions of the system.

This idea raises the basic question of continuing support of older versions of the software for those who do not wish or cannot afford to purchase new versions. Do you think this is important? Are there many users sticking with the older versions? In other product lines DEC has already been forced to deal with this problem because new versions of the software cannot run on older machines (i.e., RSTS-11).

WORDS.RA

Tom McIntyre sent along his version of a FORTRAN IV callable routine to access 12 bit data. It works just like the standard library routines "CGET" and "CPUT" except that it operates on 12 bit data words rather than 6 bit characters. A copy will be attached to this Newsletter. I would like to know if anyone has

success in running this routine on a non-FPP configuration (Tom has an FPP-12). I am not quite sure if the run time system for non-FPP systems supports some of D mode instructions used.

This version is probably more attractive than my "WORDOP" version mentioned in Newsletter No. 12 if you have an FPP.

"WHETSTONE" BENCHMARK TESTS

Several members of the SIG have helped me run a benchmark program on several different hardware and language configurations. The program is said to be the one that produced the famous "Whetstone" measurements on a wide variety of systems that have received considerable publicity. I will attach a summary of the results we have measured and a selection of the published results which are advertized as having been run against the same test. The test seems to be a reasonable exercise of FORTRAN and it has been designed to try to minimize compiler optimizations such as removing static calculations from loops. The translations to other languages are by no means directly comparable but they give some sort of minimal comparison of "number crunching" speeds.

WHETSTONE BENCHMARK REPORT

MEASURED RESULTS

MACHINE/CONFIGURATION	LANGUAGE/OPTIONS	PREC.	SPEED
PDP-8I	OS/8 FORTRAN IV	2^23	2.6
PDP-8I	OS/8 FORTRAN IV /N/Q	2^23	2.9
PDP-8E	OS/8 FORTRAN IV	2^23	3.15
PDP-8E/KE8-E EAE	OS/8 FORTRAN IV	2^23	4.67
PDP-8E/KE8-E EAE	OS/8 FORTRAN IV /N	2^23	5.13
PDP-12/FPP-12 FLOATING POINT PROCESSOR	OS/8 FORTRAN IV	2^23	34.5
PDP-8I	OS/8 FORTRAN II	2^27	2.4
PDP-8E	OS/8 FORTRAN II	2^27	2.82
PDP-8E/KE8-E EAE	OS/8 FORTRAN II /BILL KAUFMANS MODIFIED LIB8.RL FOR KE8-E	2^27	5.43
PDP-8I	OS/8 BASIC	2^23	3.1
PDP-8I	PFOCAL (OMSI)	2^35	0.34
PDP-8I	PFOCAL (OMSI)	2^23	0.37
PDP-8I	U/W-FOCAL	2^35	0.40
PDP-8I	U/W-FOCAL (JIM VAN ZEE)	2^23	0.43
PDP-8I	U/W-FOCAL 20K-V2	2^35	0.49
PDP-8E/KE8-E EAE	U/W-FOCAL	2^35	0.76
PDP-11/45, 64K CORE FPP	MOP FORTRAN	2^55	62.10
PDP-11/45, 64K CORE, FPP	RT-11 FORTRAN IV	[illegible]55	65.96
PDP-11/45, 64K CORE, FPP	FORTRAN IV PLUS	[illegible]55	169.86
IBM 360/75	FORTRAN G	?	518.6
IBM 370/158	FORTRAN H EXT. LEV. 2.1 NO OPTIMIZE VSII V1.7 HASP	SINGLE	650
		DOUBLE	475

PUBLISHED DATA

MACHINE/CONFIGURATION	LANGUAGE/OPTIONS	PREC.	SPEED
PDP-11/45 ??	??	2^23	77
DECSYSTEM-10 KA	FORTRAN	2^27	194
		2^54	56
DECSYSTEM 10 KI	FORTRAN	2^27	500
		2^54	250
NOVA 840 MMPU, FPPU, MID, 64K	FORTRAN 5 (3.01)	16^6	71
		16^14	56
IBM 360/65	FORTRAN G	16^6	430
		16^14	321
IBM 360/65	FORTRAN H OPT 2	16^6	521
		16^14	421
CDC 7600	FTN OPT-2	2^48	8333

Special thanks to Mark Lewis, FAA, Bill Kaufman, Mobil R & D Corp., Jim van Zee, University of Seattle and Tom McIntyre, West Virginia University Medical Center for helping run these benchmarks.

STANDARDIZATION OF IOT CODES FOR TIME-SHARING AND FOREGROUND/BACKGROUND SYSTEMS SUBMITTED BY W. F. HAYGOOD, JR.

The following is submitted as a proposal for a standardization of the IOT codes used in PDP-8 Foreground/Background and time-sharing systems. Each of the following pages is shown with 200, IOT codes and a proposed use for many of them. Where there is no information at the bottom of a page regarding a specific mnemonic, that mnemonic has approximately the same meaning as the corresponding EDUSYSTEM 50 mnemonic. For the additional mnemonics that I am proposing, explanations are given. I strongly feel that some of the EDUSYSTEM 50 IOT's should be accepted as a standard feature on all OS/8-oriented time-sharing systems. Among these are: KSB, SBC, UND, DUP, TOD, RCR, DATE, SYN, STM, SRA, TSS, SSW, SEA, ASD and REL.

By the way, it seems natural that PDP-8 time-sharing systems in the near future will be centered around making OS/8 available to each time-shared user. This proposal is strongly influenced by this way of thinking.

Since we have only 1000, IOT codes at our disposal, I suggest that we carefully consider which codes will best serve various purposes.

To get a duscussion going, I propose the following:

IOT Codes:	PROPOSED USAGE:
6000-6077	With few exceptions, these codes should be reserved for software simulation of the actual hardware codes.
6100-6177	No thoughts on these codes at the present time.
6200-6277	Except for CKS, RDF, RIF, and the field changing codes, these codes should be reserved for use *by* the time-sharing executive system for communication with the user program with or without the user's knowledge (highly implementation dependent, of course).
6300-6377	No thoughts on these codes at the present time.
6400-6577	EDUSYSTEM 50 has made use of some of these codes to enable user programs to obtain information from the executive system itself. This usage should be continued with these specific codes termed *executive requests*. Many services can be added using these codes to assist the user such as a floating point package. The use of some of these services should, of course, be an additional charge to the user.
6600-6777	It seems that most of the new peripherals DEC is making use IOT's in this range. In keeping with the philosophy that each user should feel that he is using a stand-alone system, these codes should be reserved for the software simulation of the actual hardware codes.

To keep things in *general* agreement with DEC's hardware IOT practices, we should consider such things as:

"SKIP ON FLAG" type IOT's should end in "1"
"CLEAR FLAG" type IOT's should end in "2"
"DO OPERATION" type IOT's should end in "4" or "6"

MINIMUM HARDWARE REQUIRED:

Any OS/8 or OS/12 computer configuration with one of the following groups of laboratory peripherals.

1. LAB-8/E, PDP-8/E

DK8-ES	Real Time Clock and Schmitt Triggers
AD8-ES	Analogue to Digital Converter
*AM8-EA	Multiplexor
VC8-E	Display Control
*DR8-EA	Digital I/O

2. AX08, PDP-8 family

AX08	Combined ADC/Scope Control RC and Crystal Clocks
*Option XR	Digital Outputs, Contingency Inputs
*Option XM	Additional Channels of Analogue Input

3. PDP-12

KW12-A	Real Time Programmable Clock
AD12	Analogue to Digital Converter and 16 Channel Multiplexor
*AM12	Additional 16 Channel Multiplexor
VC12	LINC Scope Control
VR12	Oscilloscope
*KD12	Digital Output Relays

* These components are optional. They are supported by the system but are not essential for a satisfactory performance.

NOTE - An oscilloscope of any type should be considered an essential component for any laboratory system.

OPTIONAL HARDWARE SUPPORTED:

Any other devices supported by OS/8 BASIC - such as: Additional memory, line printer, disk, PDP-8/E EAE, etc.

PREREQUISITE SOFTWARE:

OS/8 V3
OS/8 BASIC V3

Mail to:

The Small Computer Fund
Department of Pharmacology & Therapeutics
University of Manitoba Faculty of Medicine
770 Bannatyne Ave.
Winnipeg, Manitoba
R3E 0W3
Canada

For further information call:

Stan Vivian
(204) 786-3642

EXECUTIVE REQUESTS:

6400	KSB	6440	ASD	6500		6540	FSP
6401	SBC	6441		6501		6541	FAD
6402	DUP	6442	REL	6502		6542	FSB
6403	UND	6443		6503		6543	FMP
6404		6444		6504		6544	FDV
6405	CLS	6445		6505		6545	FGET
6406		6446		6506		6546	FPUT
6407		6447		6507		6547	FNOR
6410	SIZE	6450		6510		6550	
6411	URT	6451		6511		6551	FPINT
6412	TOD	6452		6512		6552	FPICL
6413	RCR	6453		6513		6553	FPCOM
6414	DATE	6454		6514		6554	FPHLT
6415	SYN	6455		6515		6555	FPST
6416	STM	6456		6516		6556	FPRST
6417	SRA	6457		6517		6557	FPIST
6420	TSS	6460		6520		6560	
6421	USE	6461		6521		6561	
6422	CON	6462		6522		6562	
6423		6463		6523		6563	
6424		6464		6524		6564	
6425		6465		6525		6565	
6426		6466		6526		6566	
6427		6467		6527		6567	
6430	SSW	6470		6530		6570	
6431	SEA	6471		6531		6571	
6432		6472		6532		6572	
6433		6473		6533		6573	
6434		6474		6534		6574	
6435		6475		6535		6575	
6436		6476	OS8	6536		6576	
6437		6477		6537		6577	

THE CODES 6400-6577 SHOULD BE RESERVED FOR "EXECUTIVE REQUESTS".
SEE FIRST PAGE FOR MORE DETAILS ON EXECUTIVE REQUESTS.

IOT: EXECUTIVE SYSTEM ACTION:

OS8 RESTORES RESIDENT OS/8 TO USER CORE AND SETS RE-START ADDRESS TO USER RELATIVE 07600 (ALLOWS USER TO USE ALL HIS CORE AND LATER RESTORE OS/8 RESIDENT PORTION).

CODES 6540-6547 SHOULD BE RESERVED FOR USE OF A FLOATING POINT PACKAGE. SEE DETAILS ON NEXT PAGE

CODES 6551-6557 SHOULD BE RESERVED FOR SYSTEMS HAVING THE FPP-12 HARDWARE FLOATING POINT PROCESSOR. SOME IMPLEMENTORS MAY DESIRE TO SIMULATE THIS HARDWARE.

CODES 6540-6547 CALL FOR FLOATING POINT OPERATIONS TO BE PERFORMED BY THE EXECUTIVE SYSTEM. THE USER'S OPERAND ADDRESS WOULD BE SPECIFIED BY THE LOCATION POINTED TO BY THE CONTENT OF THE USER PC AND INSTRUCTION FIELD. THE USER DATA FIELD WOULD INDICATE THE RELATIVE DATA FIELD LOCATION OF THE OPERAND. "FSP" WOULD BE A SPECIAL EXIT LIKE DEC'S FLOATING POINT PACKAGES EXCEPT THAT THE SUBROUTINE DESIRED (SQROOT, SINE, ETC) WOULD BE SPECIFIED IN THE LOCATION POINTED TO BY THE USER PC AND INSTRUCTION FIELD. THE FORMATS OF THE PACKAGE WOULD BE IMPLEMENTATION DEPENDENT. THE USER FLOATING ACCUMULATOR WOULD BE IN RELATIVE USER ADDRESS 00044-00047. ADVISABLE ONLY FOR SYSTEMS WITH EAE.

TYPICAL CALLING SEQUENCE:

```
        CDF 10          /OPERAND IN RELATIVE FIELD 1
        FDV             /FL DIV MY FAC BY OPERAND
           OPRAND
        RETURN IS HERE WITH IF AND DF AS BEFORE CALL
```

SUGGESTED FLOATING POINT DATA FORMAT:
TO OBTAIN MAXIMUM ACCURACY FROM A 3-WORD FLOATING POINT NUMBER WITH REASONABLE LIMIT TO EXPONENT SIZE, I PROPOSE THE FOLLOWING:

	BIT(S)	USE
WORD 1:	0	SIGN OF MANTISSA
	1-5	EXPONENT BIASED AT 200(8)
	9-11	HIGH ORDER MANTISSA IN SIGN-MAGNITUDE FORMAT WITH MSB IN BIT 8 POSITION (HIDDEN BIT)
WORD 2:	0-11	MIDDLE ORDER MANTISSA
WORD 3:	0-11	LOW ORDER MANTISSA

SOME EXAMPLES:

	HIDDEN BIT	24-BIT FPP
THE NUMBER "1" WOULD BE:	2010	0001
	0000	0000
	0000	0000
THE NUMBER "-PI" WOULD BE:	6024	0002
	4417	4567
	6652	4023

THE MANTISSA ALWAYS REMAINS A POSITIVE QUANTITY WITH BIT 0 OF WORD 1 INDICATING THE TRUE SIGN. THE USE OF THE "HIDDEN" BIT GIVES AN ADDITIONAL 4 BITS OF ACCURACY OVER THE 24-BIT FPP.

6600	6640	6700	6740
6601	6641	6701	6741 DSKP
6602	6642	6702	6742 DCLR
6603	6643	6703	6743 DLAG
6604	6644	6704	6744 DLCA
6605	6645	6705	6745 DRST
6606	6646	6706	6746 DLDC
6607	6647	6707	6747
6610	6650	6710	6750
6611	6651	6711	6751
6612	6652	6712	6752
6613	6653	6713	6753
6614	6654	6714	6754
6615	6655	6715	6755
6616	6656	6716	6756
6617	6657	6717	6757
6620	6660	6720	6760
6621	6661 LPSF	6721	6761 DTRA
6622	6662 LPCF	6722	6762 DTCA
6623	6663	6723	6763
6624	6664 LPPC	6724	6764 DTXA
6625	6665	6725	6765
6626	6666 LPLS	6726	6766 DTLA
6627	6667	6727	6767
6630	6670	6730	6770
6631	6671	6731	6771 DTSF
6632	6672	6732	6772 DTRB
6633	6673	6733	6773
6634	6674	6734	6774 DTXB
6635	6675	6735	6775
6636	6676	6736	6776
6637	6677	6737	6777

IN KEEPING WITH THE PHILOSOPHY THAT EACH USER SHOULD FEEL AS THOUGH HE IS USING A STAND-ALONE SYSTEM, THESE CODES SHOULD BE RESERVED FOR USE BY THE IMPLEMENTOR FOR SOFTWARE SIMULATION OF THE SAME HARDWARE CODES DETERMINED BY HIS HARDWARE CONFIGURATION.

FOR EXAMPLE, THE ABOVE CODES WOULD BE USED IN A SYSTEM WITH AN LS8-F LINE PRINTER, AN RK8-E DISK DRIVE, AND A TC08 DECTAPE.

digital

SOFTWARE PERFORMANCE REPORT

FIELD #:	SPR #:
FOR DEC USE ONLY	

24015

SYSTEM PROGRAM AND VERSION (OR DOCUMENT)	MONITOR AND VERSION	DATE
FORTRAN IV v II (3.03)	OS/8 III	8/26/75

NAME: Norman R. Dotti
FIRM: Kemper Corporation
ADDRESS: Long Grove, Illinois 60049
ZIP

DEC OFFICE: Rolling Meadows

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

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

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

SUBMITTED BY: Norman R. Dotti PHONE: (312) 540-2033

LIST ATTACHMENTS: Source Code and Output

CPU TYPE	SERIAL NO.	SYSTEM DEVICE	MEMORY SIZE	DISTRIBUTION MEDIUM
8E	6278	TD8e and Foreign Disk	32K	DECtape

PROBLEM: DATA Statement doesn't work as advertised.

I've found two problems with the DATA Statement, one minor, one major. The first involves combinations of variable and value lists. Page 8-112 of OS/8 Handbook shows the "correct" form, but this doesn't work. It seems that "commas" are needed to separate a variable from the preceding value. See Example, 1, attached.

Second problem is much more complicated. When working with DATA ARRAYS, the values aren't always being put in the correct elements in the arrays. After trying different things, it seems to me that the problem is in the initializing of the array by the DATA Statement, not with the look-up which pulls the values from the arrays.

Page 2 shows a small program which seems OK; wrote it after problem came up. Pages 3 and 4 show a program and BLOCK DATA subprogram which should output array elements; results are not correct.

Page 5 shows part (cut-down as part of testing) of a program that defines an array, and then dumps it; it doesn't work correctly!

I've spent some time with Frank Nicodem, DEC, Rolling Meadows, on the problem, and he has copies of all tests I've run.

CC: Frank Nicodem, Software Services, Rolling Meadows, IL
Dave Ferrarini, Lab Products, Marlboro, MA

SOFTWARE COMMUNICATIONS USE ONLY

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

SOFTWARE COMMUNICATIONS

Laboratory BASIC V4 July 1975

Laboratory BASIC is a laboratory oriented major extension to OS/8 BASIC. It is intended primarily for the high level language programmer, the investigator who wants to do his own programming, the experienced programmer who wants to spend a minimum of time in program development - without excessive execution time penalty, or for the graduate student with his first encounter with laboratory computing.

Problems best suited to Laboratory BASIC are "event" related problems where a series of events of relatively constant time course are to be acquired, averaged, smoothed, stored on and retrieved from mass storage, examined for maxima-minima, times to peak, etc. Examples include: Muscle contraction and/or relaxation phenomenon, conduction characteristics in isolated cortex slabs, and EKG analysis.

Major features of Laboratory BASIC are:

- Standard Simple High Level Language All the features of BASIC and the extensions of OS/8 BASIC are available.
- Device Independence Laboratory BASIC will run on any OS/8 or OS/12 system with laboratory peripherals and for which a 1 or 2 page handler can be written.
- Excellent Core Efficiency The laboratory overlays to OS/8 BASIC consist of 5 segments, only one of which is core resident at a time. The segments are dynamically loaded without user intervention or directory lookups.
- Trace Mode Data Storage Sampled data from a particular channel is stored in contiguous locations in the buffer to facilitate examination and analysis.
- Random Data Access Acquired events may be transferred to or from any mass storage file in random order.
- Continuous Data Transfer The whole data buffer is transferred to or from mass storage in a single operation. Thus, on DECtape a single tape motion can transfer the entire data buffer.
- Large Data Buffer The data buffer can contain any multiple of 256 data points up to 2048 in core.

- Overlay Calls may be mixed Calls to the standard arithmetic, string or file functions may be intermixed with any laboratory function calls.
- Random Access to Floating Point Data There is random access to the data in numerical files providing a virtual file capability.
- Analytical Functions There are functions to rapidly find minima-maxima, test for data continuity, or smooth the data with an N-point running average.
- Multiple-Event Averaging Multiple events may be averaged together in a floating point array buffer at high speed.

Laboratory BASIC's Functions

INI - Establish buffer size, load handler.

VPP - Put a floating point no. randomly into any numerical file.

VPG - Get a floating point no. randomly from any numerical file.

PXY - Put a pair of coordinates randomly into a buffer reserved for 2-coordinate display.

DXY - Display all, or a segment of, the 2-coordinate buffer.

CLK - Set the clock or wait for the clock or Schmitt triggers.

SAM - Burst sample at preset clock rate, any no. of channels, Schmitt wait optional. Display while sampling.

DIS - Display all, or a segment of, the sampling buffer.

MAS - Transfer the entire sampling buffer randomly to or from a mass storage file.

PUT - Put a floating point no. from 0.0 to 1.0 into the sampling buffer as a suitable integer.

GET - Retrieve sampled data from the sampling buffer as a floating point no. from 0.0 to 1.0.

DIG - Read the switch register or digital input register, or, set the digital output register.

MAX - Find location of maximum or minimum value within a selected range of the sampling buffer.

CON - Examine a selected segment of the sampling buffer for data continuity and report position of outliers.

RAV - Perform an N-point running average on the data within a selected range of the sampling buffer.

AVR - Ensemble averaging function. Adds or subtracts a selected range of the sampling buffer to or from the averaging buffer. Multiplies or divides by a constant and transfers between buffers where appropriate.

6000	6040 SAS	6100	6140
6001	6041 TSF	6101	6141
6002	6042 TCF	6102	6142
6003	6043	6103	6143
6004	6044 TPC	6104	6144
6005	6045 TSK	6105	6145
6006 SGT	6046 TLS	6106	6146
6007 CAF	6047	6107	6147
6010 RRS	6050	6110	6150
6011 RSF	6051	6111	6151
6012 RRB	6052	6112	6152
6013	6053	6113	6153
6014 RFC	6054	6114	6154
6015	6055	6115	6155
6016 RRC	6056	6116	6156
6017	6057	6117	6157
6020 PST	6060	6120	6160
6021 PSF	6061	6121	6161
6022 PCF	6062	6122	6162
6023 PCB	6063	6123	6163
6024 PPC	6064	6124	6164
6025	6065	6125	6165
6026 PLS	6066	6126	6166
6027	6067	6127	6167
6030 KSR	6070	6130	6170
6031 KSF	6071	6131	6171
6032 KCC	6072	6132	6172
6033	6073	6133	6173
6034 KRS	6074	6134	6174
6035	6075	6135	6175
6036 KRB	6076	6136	6176
6037	6077	6137	6177

IOT: EXECUTIVE SYSTEM ACTION:
SGT SKIP ON "GT" FLAG (FOR SYSTEMS WITH EAE)
CAF CLEAR AC, LINK, GT FLAG, AND SET EAE TO MODE A
RFC CLEAR READER CORE BUFFER, START READER
RRC GET A CHAR FROM READER CORE BUFFER
PCB CLEAR PUNCH CORE BUFFER
KCC CLEAR AC
KRB CLEAR AC, READ KEYBOARD
TSK SKIP ON KEYBOARD/PRINTER FLAG

CODES 6000-6077 SEEM PRETTY MUCH DEDICATED TO THE SOFTWARE SIMULATION OF THE SAME BASIC HARDWARE CODES. PERHAPS THOSE UNUSED IN THIS RANGE SHOULD BE RESERVED FOR FUTURE DEVELOPMENTS BY DEC FOR PERIPHERAL DEVICES WHICH MAY USE THESE CODES.

I HAVE NO STRONG FEELINGS REGARDING THE USE OF THE CODES 6100-6177.

6200 CKS	6240	6300	6340
6201 CDF 00	6241 CDF 40	6301	6341
6202 CIF 00	6242 CIF 40	6302	6342
6203 CDI 00	6243 CDI 40	6303	6343
6204	6244	6304	6344
6205	6245	6305	6345
6206	6246	6306	6346
6207	6247	6307	6347
6210	6250	6310	6350
6211 CDF 10	6251 CDF 50	6311	6351
6212 CIF 10	6252 CIF 50	6312	6352
6213 CDI 10	6253 CDI 50	6313	6353
6214 RDF	6254	6314	6354
6215	6255	6315	6355
6216	6256	6316	6356
6217	6257	6317	6357
6220	6260	6320	6360
6221 CDF 20	6261 CDF 60	6321	6361
6222 CIF 20	6262 CIF 60	6322	6362
6223 CDI 20	6263 CDI 60	6323	6363
6224 RIF	6264	6324	6364
6225	6265	6325	6365
6226	6266	6326	6366
6227	6267	6327	6367
6230	6270	6330	6370
6231 CDF 30	6271 CDF 70	6331	6371
6232 CIF 30	6272 CIF 70	6332	6372
6233 CDI 30	6273 CDI 70	6333	6373
6234	6274	6334	6374
6235	6275	6335	6375
6236	6276	6336	6376
6237	6277	6337	6377

CODES IN THE RANGE 6200-6277 (EXCEPT CKS, RDF, RIF, AND THE FIELD CHANGING CODES) SHOULD BE RESERVED FOR USE BY THE TIME-SHARING EXECUTIVE SYSTEM ITSELF (WHICH WOULD BE HIGHLY IMPLEMENTATION DEPENDENT).

I HAVE NO STRONG FEELINGS REGARDING THE USE OF CODES IN THE RANGE 6300-6377.

```
/FORTRAN SUBROUTINE TO FETCH AND PUT 12 BIT WORDS IN AN ARRAY
/CALLING CONVENTION IS THE SAME AS THE STANDARD LIBRARY ROUTINES
/CGET AND CPUT EXCEPT 12 BIT "BYTES" ARE TRANSFERRED.

/SMALL COMPUTER LAB
/DEPARTMENT OF PHYSIOLOGY AND BIOPHYSICS
/WEST VIRGINIA UNIVERSITY MEDICAL CENTER
/MORGANTOWN, WEST VIRGINIA 26506
/
/
/MAY 1, 1974
/
/
        SECT    WORDS
        ENTRY   WGET
        ENTRY   WPUT
        TEXT    +WORDS+
WORDXR, SETX    XRWORD  /SET INDEX REG TO ADDRESS OF XRWORD
        SETB    BPWORD  /SET BASE REG TO ADDRESS OF XRWORD
BPWORD, F 0,
XRWORD, F 0,
FROM,   F 0,            /CONTAINS VALUE OF WORD, THIRD ARGUMENT
NWORD,  F 0,            /POSITION OF ELEMENT IN ARRAY, SECOND ARG
POINT,  F 0,            /TRIPLICATE CURRENTLY BEING MANIPULATED
TEMP,   F 0,            /TEMPORARY STORAGE OF TRIPLICATE
        ORG     10*3+BPWORD
        FNOP
        JA      WORDXR
        0
WORDTN, JA      .
        BASE    0
START,  JA      .
        STARTD
        FLDA    10*3
        FSTA    WORDTN
        FLDA    0
        SETX    XRWORD  /SET X0 TO ADDR OF XRWORD
        SETB    BPWORD  /SET BASE REG TO ADDR OF BPWORD
        BASE    BPWORD
        LDX     1,1
        FSTA    BPWORD  /STR SAVED IN BPWORD
        FLDAX   BPWORD,1
        FSUB    KJA     /REMOVING THE JA
        FSTA    STR     /CONTAINS ADDRESS OF ARRAY, FIRST ARG
        FLDAX   BPWORD,1+
        FSTA    NWORD
        FLDAX   BPWORD,1+
        FSTA    FROM    /ADDR OF F
        STARTF
        FLDAX   NWORD
        FDIV    A
        ATX     0
        XTA     0
        FMUL    A       /TAKE THREE-
        FSTA    TEMP    /FIX IT INTEGER
        ALN     0
        STARTD
        FADD    STR
        FSTA    STR
        STARTF
        FLDAX   NWORD
        FSUB    TEMP
        FADD    FONE
        FSTA    NWORD
```

```
FONE,   F 1.0

KJA,    1036;0000
STR0,   27
STR,    0;0

WGET,   JSA     START
        FLDA STR0       /LOAD ADDRESS OF ARRAY
        FNORM           /CHANGE TO FLOATING FORMAT
        FADD    NWORD   /LOAD NUMBER OF ELEMENT
        FSUB    FOUR    /SUB 3 TO PUT ELEM. IN THE LS 12 BITS
        ALN 0           /PREPARE FOR DOUBLE PRECISION
        STARTD
        FSTA    POINT   /STORE ADDR OF ELEMENT & 2 PREV.
        STARTF
        FLDAX   POINT   /LOAD 3 ELEMENTS
        FSTA TEMP       /STORE TEMPORARILY
        SETX    TEMP    /PARALLELS INDEX REG ADDRESS
        XTA     2       /RETURN ELEMENT TO FAC
        FSTAX   FROM    /STORE FETCHED 12 BIT WORD
        JA WORDTN       /RETURN TO CALLING PROGRAM

WPUT,   JSA     START
        FLDA STR0       /LOAD ADDR OF ARRAY
        FNORM           /CHANGE TO FLOATING FORMAT
        FADD    NWORD   /ADD THE POS. OF ELEM. TO ADDR. OF ARRAY
        FSUB    FOUR    /SUBTRACT FOUR FOR ADDR OF 36 BIT WORD
        ALN 0
        STARTD
        FSTA    POINT   /STORE ADDR
        STARTF
        FLDAX   POINT   /LOAD THREE ELEMENTS
        FSTA    TEMP    /STORE TEMPORARILY
        SETX    TEMP    /PARALLEL THE INDEX REG WITH TEMP
        FLDAX   FROM    /GET ELEMENT
        ATX     2       /MOVE ELEMENT TO INDEX REG
        FLDA    TEMP    /LOAD CORRECT TRIPLICATE
        FSTAX   POINT   /PLACE CORRECT TRIPLICATE INTO ARRAY
        JA WORDTN       /RETURN TO CALLING PROGRAM

A,      F 3.
FOUR,   F 4.
        END
```