



DECUS 12 BIT SPECIAL INTEREST GROUP
NEWSLETTER

November

NUMBER 31

1978

Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator - 12 Bit SIG
c/o DECUS MR2-3/E55 ..or.. Liberty Mutual Research Center
One Iron Way 71 Frankland Road
Marlboro, MA 01752 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 29-December-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.

SIG COMMITTEES AND WORKING GROUPS

Steering Committee:

Robert Hassinger - address above - (617) 435-3452

Jim Crapuchettes
Menlo Computer Associates, Inc.
P.O. Box 298
Menlo Park, Calif. 94025

Lee Nichols
E. I. Du Pont
Experimental Station
Building 357
Wilmington, DE 19898
(302) 772-3839

(415) 323-3009

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

Jonathan Lockwood
 Harris Semiconductor
 PO Box 883
 Melbourne, FL 32901
 (305) 724-7542 M.S. 54-40

Special Steering Committee Advisors:

Tom W. McIntyre Stan Rabinowitz

RTS-8 Working Group

Lee Nichols - see above

Micro-8 Working Group

Jonathan Lockwood - see above

Symposium Software Exchange Committee

Send copies of software you wish to exchange at the next U.S. symposium to the appropriate committee member for preparation:

DEctapes Russell Overbey
 DEC floppys PO Box Y Bldg. 92101-2
 AED floppys Oak Ridge National Laboratory
 Oak Ridge, TN 37830

DEctapes Earl T. Ellis, Jr.
 Magtapes USCG R & D Center
 DEC floppys Avery Pt.
 Groton, CT 06340
 (203) 445-8501 Ext. 296
 (FTS) 642-7274 Ext. 296

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

OS/8 RUNOFF V6

Many of you know the value of RUNOFF. It is available in more or less compatible versions on most of DEC's machines running under nearly all the operating systems from one source or another. A very good version has been available for OS/8 (and OS/78 of course) from the University of West Virginia Medical Center for some time. Now, at last, it has been submitted to DECUS and is available to everyone. I use RUNOFF for this newsletter as well as reports and technical papers with great success. The DECUS catalog abstract is as follows:

"RUNOFF is a program which runs under PS/8, OS/8, or DECSYSTEM-8 to facilitate the preparation of typed or printed manuscripts, such as

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

memorandums, manuals, theses, etc. The user prepares his material on any terminal, and writes it into a file using TECO, EDIT, SCROLL, or any other editor available to the user. The user includes not only textual material, but also case and formatting information. RUNOFF then takes the file and reproduces it onto the line printer, teletype, or other file to produce a final copy or final file image. It performs the formatting and case shifting as directed, and will also perform line justification, page numbering and titling, etc., as desired.

The principal benefit of such a program is that files prepared for use with it may be edited and corrected easily. Small or large amounts of material may be added or deleted, and unchanged material need not be retyped. After a set of changes, the program may be operated to produce a new copy which is properly paged and formatted. Documentation may thus be updated as necessary without requiring extensive retyping."

The 12 bit world has always needed a standard format for machine readable documentation. In the case of the PDP-11 and DECsystem-10/20 the use of RUNOFF has been a de-facto standard for years thanks to the fact that versions of RUNOFF for those systems have been available in the public domain (i.e. DECUS and/or part of DEC's software distributions). Now that an OS/8 version of RUNOFF is similarly available to everyone, I think we should move toward insisting or at least strongly requesting that DECUS submissions and other widely exchanged software include machine readable documentaion, done with RUNOFF (i.e. the RUNOFF format file and, where it will fit, the final output from RUNOFF as well).

In this way, each installation can generate as many copies of the documentaion as they need and can adjust the format for their printer, etc., (some printers will not accept lower case characters for example). They can also make modifications if needed and, when someone adds a feature or makes some other modification to the software, they can update the documentaion by simply editing the RUNOFF file.

I find it very helpful when Newsletter contributors submit their inputs as machine readable (check with me on media) files in RUNOFF format. This allows me to edit the material and incorporate it directly into the body of the Newsletter with minimum effort. This procedure, for example, is what made it possible to handle Lars Palmer's large contribution this time. Please note, however, that each version of RUNOFF has a few extra features and functions that are unique to that version. Please try to avoid them when preparing material for exchange or submission to the Newsletter. I, for example, must use either of two PDP-11 versions as well as the OS/8 version of RUNOFF from time to time, so special features like the extended HEADER command, features from specially modified versions, and odd constructions that may not be exactly compatible cause problems.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

I find the following to be a good combination for the newsletter:

```
.TABS ABSOLUTE  
.PAGE SIZE 66,72  
.RIGHT MARGIN 72  
.NO FLAG HYPHEN  
.NO FLAG INDEX  
.HEADER SOLID  
.NO JUSTIFY
```

Use of these settings seems to avoid many of the compatibility conflicts I have found between the versions I use. The "NO JUSTIFY" is due to my personal opinion that it looks better for this type of work and avoids some sticky formatting problems that come up. If you are entering the material from a terminal with upper and lower case, I find it is best to include a ".UPPER CASE" command (why RUNOFF was designed with this syntax escapes me but it works) and then to enter the material just as you want it to appear. If you are using an upper case only terminal you have to use the various flags and commands to get case control of course.

Perhaps we need some articles in the Newsletter that detail how to do many of the common formatting tasks. I find that I keep learning more techniques and tricks the more I use RUNOFF and I am sure many others have found this to be true also.

In anticipation of questions about my recommendation of RUNOFF as a standard for exchange as opposed to one of the other document formatting programs, let me say this: some of the other software appears to be very useful and in some cases unique features are offered. However, RUNOFF is the de-facto standard, with public domain support across almost all DEC systems, and OS/8 RUNOFF is the only universally available and user maintainable document formatter available for the 12 bit machines with the needed features. I cannot see encouraging submission of software to DECUS that requires the purchase of a proprietary document formatter program, assembler or other special support software. DECUS submissions and other software for exchange should require only what is provided in the package you receive plus standard DEC software (PAL8 for example). In the case of documentation, because DEC does not support any document formatting software, I will settle for RUNOFF now that it is available from DECUS.

OS/8 RUNOFF is now available as DECUS 8-880 on several media. The documentation is included in machine readable format.

FORTRAN DOUBLE PRECISION

Over the years I have received many inquires regarding DOUBLE PRECISION in OS/8 FORTRAN IV. The situation is the following: if you have the version of the FPP-12 that supports it, you can use the DOUBLE PRECISION capability of OS/8 FORTRAN IV. The new FPP-8A also has support for this mode. Systems with the other versions of the FPP-12 and those without any FPP option cannot use the DOUBLE PRECISION feature because there was not enough space in FRTS to handle simulation of the hardware feature.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

Incidentally, if you have an FPP-12, you must have the one that supports this mode if you want to use the COMPLEX feature because there is only one version of the FPP-12 support in FRTS and it requires the double length floating accumulator to do COMPLEX loads and stores. The parts of FRTS that support non-FPP systems do emulate the double length loads and stores, so, as I understand it, even though you get a warning about not having double precision support available, in the case of these configurations, COMPLEX does work.

Many times I have been asked if I know of a way around the lack of support for DOUBLE PRECISION in most configurations. In the past I had to say no. Just recently I came across something that might be of help to a very industrious user that was desperate for support of extended precision, even if it meant a lot of work. In the March 1978 (volume 4 number 1) issue of the ACM Transactions on Mathematical Software, there is an article about a package of FORTRAN subroutines called MP. The abstract reads as follows: "A collection of ANSI Standard Fortran subroutines for performing multiple-precision floating-point arithmetic and evaluating elementary and special functions is described. The subroutines are machine independent and the precision is arbitrary, subject to storage limitations. The design of the package is discussed, some of the algorithms are described, and test results are given." Information on ordering listings, card decks and/or magtapes is also given. I understand that this is a very large package with thousands and thousands of lines of code. Much of the code seems to be for computing all manner of functions to arbitrary precision however (i.e. an example in the package calls one of the functions to compute the value of Pi to 100 places). It may be that someone could get by the media conversion problems and successfully configure the package to run under OS/8 FORTRAN IV with its rather odd representations of reals and integers. If so, limited subsets of the package might fit in the space available in an OS/8 FORTRAN IV program and allow at least slow calculations to any arbitrary precision desired. If anyone decides to look into this, I would appreciate hearing about it.

NOTE FROM WILLIAM R EUBANK

Mr. Eubank writes to say that he has surplus H-219A, 8k by 12 bit core memory boards from his PDP-8A for sale due to an upgrade of the system. He also says he is looking for programmers with PDP-8 assembly language experience. His address is: Bendix, Environmental & Process Instruments Division, P O Box 831, Lewisburg, W Va 24901 (304) 647-4358.

FROM LARS PALMER

The following 30 pages or so were submitted by Lars Palmer. I have done some editing and reformatting were needed to fit the Newsletter.

Dear Bob,

- PAGE 6

I read in the Newsletter No. 29, page 12, that Fred D. Brandt had problems reading from the batch stream in FORTRAN IV. I think he forgets the fact that the record after that the last read by FORTRAN must begin with a dollar at the left margin. If this is done there is no problem whatsoever with reading from FORTRAN from the batch stream. There is, however, a small patch that has to be made to FORTRAN. I presume he has made that patch so that it works at all. The patch which has been published many times in the Newsletter is in the file mentioned below. If these precautions are taken, there is no restrictions on reading data from the batch stream. However, it is impossible to use the batch stream in a program using the USR function to access file names. This is a documented restriction but easy to forget.

In the same Newsletter, page 14, Michael E. Mazzone remarks about the bit that stops using RUN on system programs in OS/78. Undocumented except in some very small of the beat remark is the fact that a simple SET SYS OS/8 will solve the whole problem.

At the recent European DECUS conference the 12 Bit SIG as traditionally had a training seminar. At this seminar was handed out a print out containing numerous bits and pieces about the OS/8 system useful for many users. That file is reproduced here. Also handed out was listings of some central tables in the OS/8 cusps which are useful to know about if you modify the cusps. By the permission of Garry Cole, these tables are also reproduced here. (Yes Bob, I have checked with Garry and publication of these tables are ok.)

A pet project of mine is to get some kind of a macro library group working. We have tried in Europe and so far produced a bit of progress. A preliminary macrel MACRO library and relocatable library has been created and does work (primitively but usefully). It is not ready to hand to the DECUS Library but a copy has been sent to Bob Hassinger, whom I hope can distribute copies, DEC tapes or floppy (Bob, if you cannot please say so). Reasonable amount of copies (also both on DEC tape or floppy) can also be obtained from me at the address given on the front page of the Newsletter.

An attempt has been made to draw up lines for standardization of such libraries and they are shown in the pages of the print outs of the documentation which are included. On the distribution media is also a program written utilizing these libraries. It is rather a simple program. The function of it is to do the same as

```
.R PIP
*FILE<TTY:
```

but keeping in mind that the input is done by secretaries

used to high speed typing and unused to data losses due to things like power fails etc (a not unknown case in computers!), I wrote a small program which will safe guard such input. It grabs whatever file space there is on the disc, precloses the file, takes the data from the keyboard, writes it block by block to file with an EOF marker that is moved forward all the time and when it sees an EOF from the keyboard, it closes the file properly with the correct size. If a power fail occurs or if ^C or anything else is done from keyboard or from the machine the file can always be recovered by PIP or TECO in a very simple manner (remember that we run on an ETOS system having up to 10 terminals where people might be typing at high speed, the data loss due to an accidental reboot or anything of the sort can be quite high).

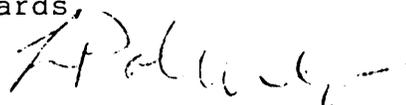
You should run this program through MACREL with the libraries available and try the following three forms of the syntax to see the effect of the listing control variables (MACLIB.RB must be on SYS:,MACLIB.MA and OS8DEF.MA on DSK:):

```
.COMP FETCH-L  
.COMP CODLIS,FETCH-L  
.COMP MACLIS,FETCH-L
```

I really hope to get comments on these files from people that are interested in getting some sort of MACREL library working group together.

I have a strong feeling that some kind of primitive MACREL librarian would be very useful. As I see it, the major need for such a librarian (oh, it could do many other things also) is to list the modules in library with a version numbers. This is really a good introductory exercise for somebody that wants to use the libraries. I do not have time at present to do a thing like that but I think many other people could have the time and I would be very happy to supply ideas to anybody that feels like attacking the problem.

Regards,



Lars Palmer
AB Haessle
Fack
S-431 20 Molndal 1
Sweden

EUROPEAN 12 BIT SIG MEMBERS

At the recent Copenhagen meeting the Steering Committee for the SIG was voted as below. Feel free to contact any of these if you need help or informaton about the 12 Bit SIG world.

Ernst Lopes Cardozo (Chairman)
Physiology Department
Vondellaan 24, Utrecht, Holland
phone: 030-88 22 21

Wim van der Mark (Symposium coordinator)
DataPlan mbH
Lindauer Str. 26
CH-Nuerensdorf, Switzerland
phone: 01-8 36 63 39

Lars Palmer (Library coordinator)
AB Haessle
Fack
431 20 Molndal 1, Sweden
phone: 031-87 01 20

Peter Stoeihrel
Inst. fur Modellstatik
Pfaffenwaldring 7
Stuttgart, Germany
phone: 0711-784 (6022/6027)

Alistair Windram
The Grassland Research Inst.
Hurley, Maidenhead, Berks, England
phone: 6288-236 314

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

The following contains a mixup of notes on OS8 useful for people interested in "advanced" OS8. No responsibility for this information is taken by anyone but as far as I know it is accurate.

OS8 blocks on disk in PIP /Y files and in core at time of assembly of the monitor.

CONTENT	PIP/Y	SYS	CORE
BOOTSTRAP	0	0	7400&6600
KM	1-4	7-12	0-1777
USR	5-7	13-15	10000-11400
HANDLERS	10-17	16-25	--
ENTER	20	26	13400
SCRATCH	21-42	27-50	---
CD	43-45	51-53	
SAVE/DATE	46-47	54-55	2000&2600
ERROR	50	56	3000
CHAIN	51	57	3400
ODT	52-55	60-63	
SPARE	56	64	
CCL REM	57	65	
12K SYSTEM	60	66	
CCL	61	67	

Here follows a hotchpotch of information ordered mainly as per program as this is the logical order when scanning through the sources:

MONITOR

JSW BITS:

0	00000-01777 NOT LOADED
1	10000-11777 NOT LOADED
2	NOT RESTARTABLE
3	DOES NOT MODIFY HIGHEST FIELD
4	CONTAINS LINK OVERLAYS
5	MAY NOT BE RUN OR GET IN OS78 SYSTEM
6	
7	
8	
9	
10	00000-01777 NOT USED
11	10000-11777 NOT USED

Scope bits are bit 4 in 17726=0.126 on SYS.

OS78 bits are bit 4 in 07771=0.371 _11.56 on SYS.

If the os78 bit is set the monitor responds "BAD CORE IMAGE" to any attempt to RUN, GET or R an system program. Simply type "SET SYS OS8" to clear this block.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

OS8 date is in 17666(as per software support) and in 07777 2-5(multiply these bits by 8 and add to year). Note that OS8 only supports 8 calendar years backwards in files !!

The following is a partial list of critical locations in the monitor changing them is to bring sorrow over your system:
 Location(as per source) presumed by:

200-203	BATCH
330	BATCH
431	CCL
435-437	CCL
1000	BATCH
1224-1236	SET /FOR SCOPE
1313-1317	SET "
1357	CCL

Many more locations are internally within the monitor presumed to be at a certain place.

N.B. when a program that does not use loc 10000-11777 is started (by RUN START or CHAIN) so must USR be in core and there is no need to call it in (used by several cusps eg PIP)

HANDLERS

Note that the KL8E source contains many kludges used by the SET program. These are specified in the latest release of the source. Read them before modifying KL8E.

The PT8E handler confuses many people by a note about 'the main assembly' this is a remnant from the PS8 time which only a few of us oldtimers know about. Just ignore it.

A very useful handler is BAT. It allows you to read from the batch input stream into your program.

CCL

CCL is a very fine program and the first that anyone interested in modifying OS8 should look at. I have here collected various bits of useful information:

- 1) The format of the CCL main table is changed in an undocumented way ugh DEC the table entries for the default switches is now in the form location;value;.....;0.
 The reason for this is the possibility to default a width in DIRECT.
- 2) Beside the main table CCL contains several useful table features that are not as well documented:
 - A) The list of default extensions is in the form:
 Extension (eg FT); Subroutine address.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

The subroutine is called before decoding to set the default processor but could do many other things.

- B) The table of CCL options (the - options) is in the form switch value;address;. ;the address points to a data block the first word of which is the address to a subroutine to be called (usually to set the output handler name). The second entry in the block often contains data used by the subroutine.
- 3) CRITICAL KLUDGES:
- A) The first two entries in the tables of commands must be as is. They are presumed by TECO and SET
- B) Location 12400 must contain COLWRD. It is presumed by SET.
- 4) I don't like the DELETE command it's too dangerous. I always force /Q on the command in the CCL table.
- 5) If you use FORTRAN heavily I suggest removing the default altmode in execute (what a lovely command by the way). Its easier to use files in that way.

The following commands are useful to place into CCL:

RNO calls RUNOFF for this printout
FUT calls FUTIL
DUMP calls OCOMP (maybe the DUMP handler is enough ?)
SORT should call SORT but is difficult to implement

If you modify CCL don't forget the COMPARE command !

RESORC

The major RESOURC tables are reproduced in appendix 1

SET

The SET program is really THE kludge of OS8. It is very dependent on the form of the MONITOR, of CCL and of several handlers. See specially KL8E. If you haven't tried it yet try "SET TTY GAG" (it's quite harmless).

PIP

Note that PIP contains a bad bug. The /Y logic is bad. This has been reported.

FOTP

There is a problem with FOTP trying to copy a device with "too many " files. Its been reported and mentioned in the newsletter. The problem is not really in FOTP but in USR but it is usually seen as a "MONITOR

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

OVERFLOW in FOTP.

FUTIL

FUTIL is a beautiful program and has almost entirely replaced EPIC which I never used anyway.

DIRECT and COREMAP

See under DECUS programs below

SRCCOM

To change SRCCOM'S test logic for comments, i.e. to get it not to compare FORTRAN comments:

Find a 7521 in field 1 (in my version (4) it is at 10074 change it to -"c or whatever have you.

A much better implementation would be /F meaning don't compare FORTRAN comments but there is very little room to patch in such a thing.

EDIT and TECO

At our installation we always use TECO, never EDIT largely because of the large compatibility between DEC operating systems. That TECO should be much harder to learn than EDIT is false. We have several secretaries using TECO and RUNOFF that never have seen a computer before much less thought that they would use one in their daily work. The macro capacity of TECO and the MUNG command are very useful. DEC has just released a very neat little booklet showing all the TECOS for the various DEC machines. A very fine little document, get it if you use TECO. (I understand that thanks for it should go to S.R. whom some of you know.)

LANGUAGE processors

PAL8 and CREF

Note that PAL8 in an environment where corefields are swapped might run more efficiently if the old symbol table is used. To modify the number of lines/page in assembler listings the locations are: PAL8 1104 CREF 77&100

MACREL

MACREL and LINK are very fine programs. An attempt to produce a macro and relocatable library has been started.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

F IV

COMPILER The /M option to the compiler (only list lines in error)
 <thanks to Alister Windram> is extremely useful.

FRTS Two very useful tables in FRTS are appendix. To make FRTS
 default another unit on its default unit numbers change in
 these tables.

To allow FRTS to use a KL8 type device as lineprinter:
 Location 254 from 6665 to 6655
 location 511 from 6663 to 7000

To use the BAT handler with FRTS (very useful) change:
 12744 from 4600 to 2000
 12745 from 3000 to 200
 but note that this makes your machine about 2 K smaller under
 BATCH or in an TD8E system (the 12K version).
 Also note that the line after the last read by FIV must start
 with \$ <dollar>, a \$JOB is useful.

A very useful patch to FRTS allows a program to trap illegal input and
 so avoid "INPUT ERROR" aborting execution

Several very useful subroutines to FIV are available:

Name	Does	From
USR	calls USR (allows dynamic names)	DECUS
MODE8	call PDP8 routines	DECUS
FILSIZ	find size of open files	LP
OCHAR	send 1 char any ASCII code	LP
ICHAR	read " "	LP
CMDOPT	read command decoder switches	LP
VT55	VT55 support	DECUS

NON DEC CUSPS

DIRECT V5G

DIRECT V5 is an extremely fine program. Allows you to sort the
 directory listings by name, extension or date.

RUNOFF

RUNOFF is a text formatting program available on PDP8'S, PDP11'S and the
 PDP10. It is a very useful tool for program documentation (what an ugly
 word). It is available from SMALL COMPUTER LAB, USA but according to
 reports it will come in the DECUS library. (Note: Now available as
 DECUS 8-880. R.H.)

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

SORT MERGE & EXTRACT

Available under the same terms as RUNOFF. Very useful programs. (Note: Contact Small Computer Lab, Dept of Physiology and Biophysics, West Virginia University Medical Center, Morgantown, West Virginia 26506, USA. R.H.)

CMU

There exists a DEC10 (!) program called TMSR that runs in the PDP8 and makes it behave as a terminal to the 10. CMU is a very upgraded version of TMSR that also has file transfer capability.

COREMAP

COREMAP replaces BITMAP. It gives the same information but in a much neater fashion.

COMPAF

A quite nice program. Checks all files on two devices with the same name to see if they are the same.

DCP

A very powerful reverse assembler originating in HOLLAND. It will find literals and assign names to locations on the first pass through a file.

FORMAT OF HELP FILE ALMOST CORRECT !

This program requires an input file to run. For OS/8 the file should be named "HELP8.HL". For OS/78 the name should be "HELP78.HL". The format of the file is:

```
SP>CR>LF>
&command name
&additional command names(if any)
text of help info
&next command name
text of next help info
etc.
```

Note: to speed up processing a hashing algorithm is used which requires that when multiple commands reference a single subfile, the subfile must appear in the help file alphabetically according to the 1st command string letter closest to Z. That is, subfiles must be arranged alphabetically according to the highest lettered 1st character of the command.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

RESOURC TABLES FROM "14 OS/8 RESOURCES PROGRAM"

12664	0000	PUTPT1,	0		
12665	0000	PUTPT2,	0		
12666	0000	PUTKNT,	0		
12667	7001	ZLQP,	IAC	/23	
12670	7001	PLAT4,	IAC	/22	
12671	7001	PLAT3,	IAC	/21	
12672	7001	PLAT2,	IAC	/20	
12673	7001	PLAT1,	IAC	/17	
12674	7001	ZXTRA,	IAC	/16	
12675	7001	ZKL8E,	IAC	/15	
12676	7001	ZRK8E,	IAC	/14	
12677	7001	ZRK8,	IAC	/13	
12700	7001	ZASR,	IAC	/12	
12701	7001	ZTA8,	IAC	/11	
12702	7001	ZTD8,	IAC	/10	
12703	7001	ZL645,	IAC	/7	
12704	7001	ZLPSV,	IAC	/6	
12705	7001	ZLV8E,	IAC	/5	
12706	7001	Z026,	IAC	/4	
12707	7001	Z029,	IAC	/3	
12710	7001	ZLSPT,	IAC	/2	
12711	7001	ZPT8E,	IAC	/1	
12712	7104	NOKIND,	CLL RAL		
		/PRINTS	NAME FROM TABLE,	ENTER WITH ENTRY	IN AC
12713	1364	PRNAM,	TAD (TABASE		
12714	4763'		JMS PUTNAM		
12715	4762'		JMS UNIT		
12716	5761'		JMP KINDRET		
13126	0424	LIST1,	TEXT	/DTMTLTTDCSRKRFRX/	
13127	1524				
13130	1424				
13131	2404				
13132	0323				
13133	2213				
13134	2206				
13135	2230				
13136	0000				
		/	ZBLOCK 1		/PATCH SPACE
14435	1420	LIST2,	DEVICE LPT		
14436	2400				
14437	2424		DEVICE TTY		
14440	3100				
14441	2024		DEVICE PTR		
14442	2200				
14443	2024		DEVICE PTP		
14444	2000				
14445	0304		DEVICE CDR		

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

```

14446 2200
14447 2331      DEVICE SYS
14450 2300
14451 0423      DEVICE DSK
14452 1300
14453 0304      DEVICE CDP
14454 2000
      /
      /      DEVICE DEV
      /      DEVICE OUT
      /      DEVICE INP
14455 0201      DEVICE BAT
14456 2400
14457 1625      DEVICE NULL      /SHOULD BE IN NEXT TABLE
14460 1414
14461 1423      DEVICE LST      /V3C
14462 2400
14463 0425      DEVICE DUMP
14464 1520
14465 2314      DEVICE SLU
14466 2500
14467 1421      DEVICE LQP
14470 2000
14471 0000      ZBLOCK 10      /PATCH SPACE
14501 0000      0

```

```

      /LIST3, DEVICE NULL
      /      DEVICE TEST
      /      DEVICE LIST
      /      DEVICE DUMP      /V3C
      /      ZBLOCK 2      /PATCH SPACE
      /      0

```

/INTERESTING NOTE: 'BAT', 'FOO2' AND 'RKC6' ALL HASH OUT TO 6601
 /** WANT TO CHANGE AIW PRINTER TO SAY 'NO' ADDITIONAL INFO WORDS
 /IF THERE ARE NONE.

```

14513 4040  TABASE, 4040;4040      /0
14514 4040
14515 2024      DEVICE PT8E      /1
14516 7005
14517 1323      DEVICE KS33      /2
14520 6363
14521 6062      DEVICE 029      /3
14522 7100
14523 6062      DEVICE 026      /4
14524 6600
14525 1426      DEVICE LV8E      /5
14526 7005
14527 1420      DEVICE LPSV      /6
14530 2326
14531 1466      DEVICE L645      /7
14532 6465
14533 2404  KTD8,      DEVICE TD8      /10

```

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

14534	7000			
14535	2401	KTA8,	DEVICE TA8	/11
14536	7000			
14537	0123		DEVICE AS33	/12
14540	6363			
14541	2213		DEVICE RK01	/13
14542	6061			
14543	2213		DEVICE RK05	/14
14544	6065			
14545	1314		DEVICE KL8E	/15
14546	7005			
14547	3024		DEVICE XTRA	/16
14550	2201			
14551	4075		TEXT / =1/	/17
14552	6100			
14553	4075		TEXT / =2/	/20
14554	6200			
14555	4075		TEXT / =3/	/21
14556	6300			
14557	4075		TEXT / =4/	/22
14560	6400			
14561	1421		DEVICE LQP	/23
14562	2000			
14563	3124			
14564	3745			
14565	0033			
14566	2542			
14567	0040			
14570	2344			
14571	2200			
14572	2600			
14573	0317			
14574	1431			
14575	7667			
14576	0171			
14577	1400			
	4600			

PAGE

/FORMAT OF SYSTEM HEAD FILE

/REL BLK	CONTENTS	ABS BLK ON DEV
/0	BOOTSTRAP PAGE 0'S	0
/1-4	KEYBOARD MŌNITOR	7-12
/5-7	USR	13-15
/10-17	DEVICE HANDLERS	16-25
/20	ENTER	26
/21-42	SCRATCH BLOCKS	27-50
/43-45	COMMAND DECODER	51-53
/46-47	SAVE,DATE	54-55
/50	ERROR OVERLAY	56
/51	CHAIN OVERLAY	57
/52-55	ODT	60-63

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

/56 RESERVED FOR EXPN 64
 /57 CCL SCRATCH 65
 /60 12K TD8E HANDLER 66
 /61 CCL OVERLAY 67

0000 FIELD 0

2000 *2000

/ DEVICE LENGTH TABLE

/FORMAT OF THIS TABLE:

/1,2 DEVICE GENERALIZED NAME (CORR TO TYPE)
 /3 NEG OF LENGTH
 /4 ADDRESS OF SUBR IN FIELD 1 FOR SPECIALIZATION

02000	2424	TYPTBL, DEVICE TTY ;0000;QTTY	/0	
02001	3100			
02002	0000			
02003	2730			
02004	2024	DEVICE PTR ;0000;QPTR	/1	
02005	2200			
02006	0000			
02007	4042			
02010	2024	DEVICE PTP ;0000;QPTP	/2	
02011	2000			
02012	0000			
02013	4042			
02014	0322	DEVICE CR8E;0000;QCR8E	/3	
02015	7005			
02016	0000			
02017	4051			
02020	1420	DEVICE LPTR;0000;QLPTRX	/4	V3D
02021	2422			
02022	0000			
02023	5346			
02024	2213	DEVICE RK8 ;1520;QRK8	/5	
02025	7000			
02026	1520			
02027	2746			
02030	2206	DEVICE RF08;6001;PLAT1	/6	V3C
02031	6070			
02032	6001			
02033	2673			
02034	2206	DEVICE RF08;4002;PLAT2	/7	
02035	6070			
02036	4002			
02037	2672			
02040	2206	DEVICE RF08;2003;PLAT3	/10	
02041	6070			
02042	2003			
02043	2671			

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

02044	2206	DEVICE RF08;0004;PLAT4	/11	RF'S NOW ONLY HAVE LOGICALLY 1777 BLOCKS
02045	6070			
02046	0004			
02047	2670			
02050	0406	DEVICE DF32;7601;PLAT1	/12	
02051	6362			
02052	7601			
02053	2673			
02054	0406	DEVICE DF32;7402;PLAT2	/13	V3C
02055	6362			
02056	7402			
02057	2672			
02060	0406	DEVICE DF32;7203;PLAT3	/14	
02061	6362			
02062	7203			
02063	2671			
02064	0406	DEVICE DF32;7004;PLAT4	/15	DF'S HAVE 177 BLOCKS
02065	6362			
02066	7004			
02067	2670			
02070	2403	DEVICE TC08;6437;QTC08	/16	
02071	6070			
02072	6437			
02073	3561			
02074	1411	DEVICE LINC;6437;QLINC	/17	
02075	1603			
02076	6437			
02077	3561			
02100	2415	DEVICE TM8E;0000;0	/20	
02101	7005			
02102	0000			
02103	0000			
02104	2404	DEVICE TD8E;6437;QTD8E	/21	
02105	7005			
02106	6437			
02107	4102			
02110	0201	DEVICE BAT ;0000;0	/22	
02111	2400			
02112	0000			
02113	0000			
02114	2213	DEVICE RK8E;1520;QRK8E	/23	
02115	7005			
02116	1520			
02117	2734			
02120	1625	DEVICE NULL;0000;0	/24	
02121	1414			
02122	0000			
02123	0000			
02124	2230	DEVICE RX8E;7022;0	/25	
02125	7005			
02126	7022			
02127	0000			
02130	0000	ZBLOCK 4	/26	

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

02134	2401	DEVICE TA8E;0000;QTA8E	/27
02135	7005		
02136	0000		
02137	4122		
02140	2622	DEVICE VR12;0000;0	/30
02141	6162		
02142	0000		
02143	0000		
02144	0000	ZBLOCK 4	/31
02150	0000	ZBLOCK 4	/32
02154	0000	ZBLOCK 4	/33
02160	0000	ZBLOCK 4	/34
02164	0000	ZBLOCK 4	/35
02170	0425	DEVICE DUMP;0000;0	/36
02171	1520		
02172	0000		
02173	0000		
02174	0000	ZBLOCK 4	/37
02200	0000	ZBLOCK TYPTBL+400-.	
	2400	PAGE	

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

FRTS TABLES FROM "FORTRAN IV RUNTIME SYSTEM, V5A"

/DSRN IMAGE

00100	0000	HAND,	0	/HANDLER ENTRY POINT
00101	0000	HCODEW,	0	/HANDLER LOAD ADDR
				FIELD + IOFFLG + FORMS CTL FLG
00102	0000	BADFLD,	0	/BUFFER ADDRESS AND FIELD
00103	0000	CHRPTR,	0	/ACTUALLY A WORD POINTER
00104	0000	CHRCTR,	0	/COUNTER - RANGES FROM -3 TO -1
00105	0000	STBLK,	0	/STARTING BLOCK OF FILE
00106	0000	RELBLK,	0	/CURRENT RELATIVE BLOCK NUMBER
00107	0000	TOTBLK,	0	/LENGTH OF FILE
00110	0000	FFLAGS,	0	/FILE FLAGS:
				/BIT 0 - "HAS BEEN WRITTEN" FLAG
				/BITS 1-2 -
				FORMATTED/UNFORMATTED FLAGS
				/BIT 11 - "END-FILED" FLAG

/INPUT BUFFER - CONTAINS STARTUP CODE

04000	7572	INBUFR,	-206	/LENGTH
04001	0000		0	/INPUT LINE BUFFER
				- FIRST A LITTLE PADDING,

/RTS EXECUTION INITIALIZATION - IN INPUT BUFFER

04002	6601	FPSTRT,	6601	/CLEAR DF32 FLAG
04003	6022		PCF	/HSP FLAG
04004	6012		RRB	/HSR FLAG
04005	7600	PP7600,	7600	/CLEAR READER CHAR
04006	6135		6135	/CLEAR KW12 OR DK8-EP EVENT FLAGS
04007	7200		CLA	
04010	6132		6132	/STOP KW12 CLOCKS
04011	6134		6134	/DISABLE KW12 INTERRUPTS
04012	6530		6530	/CLEAR AD8-EA FLAGS
04013	6050		6050	/CLEAR VC8/E FLAG
04014	6500		6500	/DISABLE XY8/E INTERRUPTS
04015	7240		STA	
04016	6130		6130	/DISABLE DK8-EP INTERRUPTS
04017	7200		CLA	/LEAVE SPACE FOR ADDITIONAL CLEARS
04020	7000		NOP	
04021	7000		NOP	
04022	7000		NOP	
04023	7000		NOP	
04024	7000		NOP	
04025	7000		NOP	
04026	7000		NOP	
04027	7000		NOP	
04030	7000		NOP	
04031	7000		NOP	
04032	7000		NOP	
04033	3025		DCA	EOLSW

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

```

04034 4553 LDPROG, JMS I [FPGO /START UP FPP OR PSEUDO-FPP
04035 4157 STSWAP
04036 7000 HLTNOP, NOP /SET TO HLT IF /H SPECIFIED,
04037 5351 JMP PRTCR /SKP IF /P SPECIFIED
04040 1237 TAD .-1
04041 3234 DCA LDPROG /BYPASS LOADING ON STARTUP
04042 1320 TAD PCHWD /HLT
04043 3777 DCA I (PDPXIT+1
  
```

/OVERLAY AND DSRN TABLES

```

4204 *. -4 /FIRST ENTRY IN OVLYTB ONLY NEEDED
      TO LOAD MAIN PGM

04204 0000 OVLYTB, ZBLOCK 40 /OVERLAY TABLE

04244 0306 DSRN, PTR; ZBLOCK 10
04245 0000
04255 0270 PTP; ZBLOCK 10
04256 0000
04266 0236 LPT; ZBLOCK 10
04267 0000
04277 0320 TTY; 0;0
04300 0000
04301 0000
04302 1234 1234 /*K* PREVENT PROBLEM IN
04303 0000 ZBLOCK 5 /RWINIT INVOLVING WRITE
      /AFTER READ ON TELETYPE

04310 0000 ZBLOCK 55

04365 0000 ZBLOCK 12 /FORMAT PARENTHESIS PUSHDOWN LIST
04377 0000 FMTPDL, 0 /GUARD WORD
      4400 PAGE

13517 0306 IHTBL, PTR;PTP;LPT;TTY /INTERNAL HANDLER TABLE
  
```

DECUS 12 BIT SPECIAL INTEREST.GROUP NEWSLETTER
Number 31 - November 1978MACLIB
Documentation to MACLIB, a macro library for MACREL
LARS PALMER - June 1978

INTRODUCTION

The macro library is constructed according to the general guidelines in the MACREL manual. To make a particular macro available to the user program the sequence is:

```
.INCLUDE MACLIB  
.MCALL A,B,C           with A,B,C being macros.
```

LISTING CONTROL

Listing is controlled by 2 flag variables. By defining them in your source the listing is switched:

None defined	List only the macro call
.CODLIS defined	List the macrocall and generated inline code
.MACLIS defined	List generated code and the macro expansion

The macro library also presumes that the subroutines in LNKLIB are linked with your program and that the definitions found in MACDEF are made.

NAMING CONVENTIONS.

All macro names start with ".". Names generated by the macros that the user normally does not see start with ".." for variables and "\$." for sections.

RESTRICTIONS

The text 'changes sect' means that the macro assembles a .SECT instruction. It always resets the sect via a .SECT * but note that LITERALS (CURRENT PAGE AND PAGE ZERO) are not re-set across a macro.

<...> in the argument list means that the argument is optional and can be left out.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

DOCUMENTATION FORMAT

Name of macro(s)

Purpose of macro

Calling sequence

Notes on expansion

Required other modules

Space required

Labels generated

Special restrictions etc

If call is not listed it is simply MNAME where MNAME is the name of the macro.

OS8 COMMUNICATION MACROS

MACRO USR

Purpose of macro: Force space to be saved for USR

Space required: Loads 10000-12000

Labels generated: .U1 to .U20

MACRO .ODT

Purpose of macro: Save space for ODT in current field

Calling sequence: .ODT<,SECT>

Notes on expansion: Generates sect .\$ODT in field SECT if given, else in field of MAIN

Space required: 4-6 in current field

Special restrictions etc: Changes sect

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

MACRO .INIT

Purpose of macro: Establish initial values for some variables, must be called before .TYPES or .FATAL

Calling sequence: .INIT <HOLD>

Notes on expansion: In field of SECT or if undefined MAIN is inserted ..USR used by .USRIN. If HOLD is included HOLD is presumed to be the address of a user entry point in the current assembly (could be defined as EXTERNAL by the user), to which the program is to trap on .FATAL.

Space required: Current field 1 page 0; Field 0 possibly 6 page 0

Labels generated: .\$INIT, .\$HLD, P7600\$, C7600\$

Special restrictions etc: Changes sect

MACRO .EXIT

Purpose of macro: Exit to OS8

Notes on expansion: Expands to :
 TMSKIP (if it is defined)
 IOF;CLA;TAD (4706 ;CDF CIF 0
 DCA 7600;JMP 7600

Space required: 6

MACRO .DECODE

Purpose of macro: Calling commandecoder

Calling sequence: .DECODE <EX>

Notes on expansion: Expands to:
 SKP;JMP NOCOD
 CIF 10;JMS 7700;DECODE;X
 X is EX if defined else 5200(special mode call)

Space required: 6

Special restrictions etc: Start address is set to this point, the macro will handle a chain start.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

"USR" INTERACTION MACROS

Name of macro(s): .USRIN .USRROUT .USRCL
 Purpose of macro: Calling USR
 Calling sequence: .USRIN .USRROUT .USRCL a
 Notes on expansion: Expands to :
 .USRIN:
 CIF 10;JMS 7700;10;TAD (200;DCA ..USR
 .USRROUT:
 CIF 10;JMS I ..USR;11;TAD (7700;DCA
 ..USR
 .USRCL A:
 CIF 10;JMS I ..USR;A
 Space required: 6/6/3
 Special restrictions etc: Must be in field defined by .INIT

"BATCH" INTERACTION MACROS

Name of macro(s): .IFBTCH, .BTCHON, .BTCHOF
 Purpose of macro: Test if batch; change output to batch
 Notes on expansion: .IFBTCH skips if batch running.
 Expansion :
 CDF 10;TAD 7777;CDF .;SMA CLA
 .BTCHON switches terminal output to
 batch log
 .BTCHOF switches it back to terminal
 (output via .TYPE)
 Expansions: see listing
 Required other modules: .IFBTCH
 .BTCHON/OF: .TYPE
 Space required: .IFBTCH 5
 .BTCHON 1st call 14. Subseq calls 7
 .BTCHOF 5
 Labels generated: .BTCHON (1st call) NOBTCH\$

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

CONTROL C HANDLING MACROS

Name of macro(s)	.TRAP, .TRPON and .TRPOF
Purpose of macro	Trap ^C
Calling sequence	.TRAP A; other no arg
Notes on expansion	.TRAP must be called before the others, A is the address to be trapped to on ^C .TRPON: CDF 0;TAD (JMP ..TRP);DCA 7600;CDF . .TRPOF: restores 07600
Labels generated	.\$TRP, ..TRP

MACRO SETJSW

Purpose of macro	Set bits in JSW
Calling sequence	.SETJSW A
Notes on expansion	Expands to: CDF 0;TAD (A;CMA;AND JSW;TAD A DCA JSW;CDF .FLD;
Space required	8

VARIOUS MACROS

MACROS .CALL and .CALLZ

Purpose of macro	Simplify calls to EXTERNS
Calling sequence	.CALL<Z> A
Notes on expansion	Expands to : .EXTERN A CIF A JMS A <u>OR</u> JMS I [A
Space required	2 loc & literal

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

MACRO .IOT

Purpose of macro Create given IOT
 Calling sequence .IOT RES,DEV,IOT
 Notes on expansion An example is simplest:
 .IOT KSF2,40,KSF
 makes KSF2=6403

MACRO TYIOT

Purpose of macro Change IOT for type
 Calling sequence .TYIOT IO
 .TYIOT 66 changes output on .TYPE to
 device 66
 Required other modules .TYPE
 Space required 8
 Labels generated A\$, B\$

MACROS .FATAL and .TYPES

Purpose of macro To test condition and type error message
 Calling sequence Typical calling:
 TAD X
 SPA CLA
 .FATAL "BAD ERROR"
 .FATAL aborts to 7600
 .TYPES returns

Notes on expansion Inline 1(one) instruction is generated
 (JMS I [..TYPES). The texts and
 addresses are dumped by calling .END.
 They are generated out of code (by
 creating macros).

Required other modules Several library modules
 Space required Inline 1,out of line 2+textstring; Also
 much space is required in MACRELS macro
 storage
 Labels generated Several of the form ..TN,..RN,..MN where
 N is an octal number

Special restrictions 1).INIT must have been called from the
 same field before 1st call
 2)the routines can only be called from 1

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

field (but the library routines may
 reside in another
 3).END must be called at end to dump the
 macros

LOGICAL MACROS

Purpose of macros

Generate logical operations

.AND. a and b
 .IOR. a inclusive or b
 .BCLR. clear bits b in a
 .NOR. a nor b
 .NAND. a nand b
 .EOR. a exclusive or b

Arguments are A,R for all

Expansions:

.AND.	.IOR.	.BCLR.	.NOR.	.NAND.	.EOR.
TAD A	TAD A	TAD B	TAD A	TAD A	TAD A
AND B	CMA	CMA	CMA	AND B	AND B
	AND B	AND A	DCA T\$	CMA	CMA IAC
	TAD A		TAD B		CLL RAL
			CMA		TAD A
			AND T\$		TAD B
			SKP		
			T\$, 0		

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

LNKLIB

The library LNKLIB is constructed in such a way as to be possible to keep all the sources in one file. To this end they are collected in the file LNKLIB.TE, a TECO macro file. This file is submitted for processing by TECO, BATCH, EXPIP and MACREL by the command:

```
.MUNG LNKLIB<,LIST_OPTIONS>
```

LIST_OPTIONS is any legal CCL listing options such as -L, it can be left out if no listings are desired.

GENERAL PRINCIPLES.

The library is constructed according to the following general principles:

- 1) All user callable names start with ".", all sections start with ".\$", all global names not normally called by the user (called by other LNKLIB routines or by MACLIB macros) start with ".."
- 2) No backward references must be in the library
- 3) The format of the sources to function with the TECO macro in the library file is:

```
/# NAME.MA          (where NAME is the name of the source, the
  characters must start at the top of a page and the tab after # must
  be there)
```

The body of the source in standard MACREL format

```
/' (to signal the end of the source)
```

- 4) The general format of the documentation is (varies a bit due to special requirements of some modules):

```
Name of module (source name on compilation).
Purpose of module
Sections used
Entry points
Globals in module
Size
Routines called, externals referenced
Calling
Special features
Restrictions
```

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

Name: TYPEC

Purpose: Types the characters following the JMS, terminator is a zero.

Sections: .\$TYPCH

Entry point: .TYPCH

Size: 16 locations

Routines called: .TYPE

Calling:

```

.CALL .TYPCH
  301          /"A"
  302          /"B"
  303          /"C"
  0           /TERMINATOR
  return - AC=0

```

Restrictions: Field restricted to field of .TYPE

Name: DATE

Purpose: Print the OS8 date

Section: .\$DATPR

Entry point .DATPR

Size: 66 loc

Routines called: .PRNT2

Calling:

```

.CALL .DATPR

```

The date is printed as: 07/17/72.

In OS8 the date is stored in loc 7666, field 1:

7666 MMMDDDDYY /M=month, D=day, Y=year

Also the year offset is stored in 07777 bits 3-5

Name: PRNT2

Purpose: Print two digits in decimal

Section: .\$PRNT2

Entry point: .PRNT2

Size: 36 locations

Externals called: .TYPE

Calling:

```

TAD (VALUE
.CALL PRNT2
  return - AC=0

```

The value of the AC is printed in two digits correctly if .LT.
 99(decimal).

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

Name: DECIN

Purpose: Subroutine reads a decimal number from keyboard.

Section: .\$DECI

Entry point: .DECIN

Size: 51 locations

Routines called: .TYPE & .READ

Calling :

```
.CALL .DECIN
    return with number binary in AC
```

Rubout removes number completely.

Name: DOCTPR

Purpose: Double word octal print routine

Section: .\$DOCT

Entry point: .DOCT

Size: 51 locations

Routines called: .OCT, .TYPE

Externals referenced: .OCTFG, .OCTSC

Calling:

```
.CALL .DOCT
    hhhh          /High order number
    llll          /Low order number
    return - AC=0
```

Name: PARTY

Purpose: Parity generator. Generates odd or even parity bit (8th bit)

Section: .\$PARTY

Entry point: .PARTY

Size 30: locations

Calling:

```
TAD CHAR
.CALL .PARTY
    return with char in AC
```

Restriction: This version only generates even parity.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 31 - November 1978

Name: LIST

Purpose: List search, cross field callable. The routine matches the AC against all elements of a list. In case of a match it takes the normal return with the offset in the list in the AC. In case it encounters a 0000 in the list, it takes the error return, also with offset in AC. The list is assumed to be in field of call. In the example: element 301 has offset 0

Calling :

```

TAD (AC
.CALL .LISTS
LST-. /MARK THIS CONSTRUCTION!!!
not in LST return (not found) - AC=OFFSET IN LST
normal return (found) - AC=OFFSET IN LST
.
LST, 301;302;303;304;0 /ZERO IS TERMINATOR!!!

```

WARNING: These routines have been recoded for MACREL but not tested.

Name: BRANCH

Purpose: Relative brancher; cross field callable. This branch routine can be called from any field. It assumes the list in the field of call, and the destination addresses too. It works with relative distances and is therefore useful for runtime relocatable programs.

Calling:

```

TAD (AC
CALL, .CALL .BRANCH
LIST--1 /RELATIVE DISTANCE TO LIST
not in list return - AC=0
.
LIST, 215;CR-.
212;LF-.
377;RUB-.
0 /ZERO IS TERMINATOR !!!!

```

WARNING: These routines have been recoded for MACREL but not tested.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

Names: PUSHX, POPX

Purpose: Push and pop operators. The routines can operate on different stacks which are pointed to by the argument. Convention is that the pointer always points to an element, unless count=0, then it points to itself.

Calling:

```
TAD (AC
.CALL .PUSHX    /PUSH C(AC) ONTO STACK
  STACK
  stack full return - AC=0
  normal return - AC=0

.CALL .POPX     /FETCH ONE WORD TO AC FROM STACK
  STACK        /POINTS TO STACK
  stack empty return - AC=0
  normal return - AC=ELEMENT
```

```
STACK, -21      /-MAX SIZE OF STACK=21(8)
  0             /COUNTER OF ELEMENTS; 0=EMPTY
  .            /POINTER, SET TO CURRENT LOC.
  1            /FIRST ELEMENT
  2;3;4.....0 (LAST ELEMENT)
```

WARNING: These routines have been recoded for MACREL but not tested.

Name: DPRINT

Purpose: Decimal print with variable number of digits. Call with number to be printed in AC and # of digits to be printed following the subroutine call.

Calling:

```
TAD NUMBER
.CALL .DPRT
  2             /# OF DIGITS TO BE PRINTED (MAX=4)
```

Name: CORE

Purpose: This is taken from OS8 Support Manual. It is obviously not correct there. I hope this is correct. It works on an 8E anyway. Modified to be callable from any field.

Calling:

```
.CALL .CORE
  return - AC= core size
```

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 31 - November 1978

Name: ERROR

The module ERROR contains routines used by the .TYPES and .FATAL macros. They are not normally accessed by the user.

Name: PRSNA

Purpose: To print a string given the address. The string may be in any field (i.e. 3 fields may be involved).

Calling:

```

          TAD      XCDF (or RDF or TAD YCDF or CLA)
          .CALL    .PRSNA
              address of TXT
XDCF,    CDF     TXT (or YCDF, .FLD TXT)
  
```

As shown the field of TXT can be used in any of 2 forms as an CDF or as a field number, or if AC=0, the string is presumed to be in field of caller (= data field at call).

Name: TYPE

Purpose: Contains the common output routines. Also used in conjunction with macros in the MACREL library allows switching of output to BATCH stream or to another teletype compatible device.

Section: .\$TYPE

Entry points: .CRLF, .STRIN & .TYPE

Globals: ..IO1,..IO2,..IO3,..IO6

Size: 104 locations

References: none

Calling:

```

1)      .CALL .CRLF           /No arguments

2)      TAD ("A
          .CALL .TYPE

3)      .CALL .STRING
          MSG
          :
          :
MSG,    TEXT .....         /in same field !
  
```



THE
GEORGE
WASHINGTON
UNIVERSITY
MEDICAL CENTER

Office of Computer Assisted Education / 2300 Eye Street, N.W. / Washington, D.C. 20037 / (202) ~~328-1347~~ ⁶⁷⁶⁻²⁶⁹²

October 19, 1978

Mr. Bob Hassinger, Coordinator
12 Bit Sig
Liberty Mutual Research Center
71 Franklin Road
Hopkinton, MA 01748

Dear Mr. Hassinger,

I want to express my thanks and appreciation to everyone who responded to my request for information on PDP-8 compatible cross-assemblers for microprocessors. The response was fantastic! I have tried to respond to everyone who wrote or called but in case I missed someone I wanted to thank everyone publicly.

The responses indicated two commercial sources of cross-assemblers which run on a PDP-8, or 12, under OS/8. These are:

- 1) Sierra Digital Systems
1440 Westfield Avenue
Reno, Nevada 89509
(702) 329-9548
- 2) FBE Research Company
P.O. Box 68234
Seattle, Washington 98168

I am enclosing a copy of the materials I have received about the offerings from these firms. I also learnt of a 6800 cross-assembler written by Mr. R. J. Sand of E.I. Dupont Denemours & Company, Savannah River Laboratory, Aiken, South Carolina 29801 [(803) 725-6211 x3397].

Sincerely,

Roy A. Standing
Programmer

14-OCT-78

PAGE 37

Mr. Robert Hassinger
Coordinator - 12 Bit Sys
c/o: Liberty Mutual Research Center
71 Franklin Road
Horsington, MA 01748

Dear Bob:

I Just received my copy of the 12 Bit Newsletter (September - Number 30 - 1978) and noted your discussion on page 6 about the problems with passing the /A option to DIRECT (user version) so as to obtain alphabetized directories.

This letter is to advise PDP-8 fans that there is a temporary means of settings around the problem, as I mentioned to you last week. The method is to use the parentheses for passing the switches to DIRECT, (A) instead of /A.

Thusly:

.DIR /A Will NOT give an alphabetized directory

Whereas:

.DIR (A) Will give an alphabetized directory

For the benefit of any newcomers out there, I should mention that alphabetized directories can only be obtained by via the user enhanced version of DIRECT. The DEC version of DIRECT does not yet support this feature. Also, for those who missed the prior discussion, the problem is in CCL itself, not in DIRECT. It seems that the CCL mapping of the switches "loses" the /A switch and this may be partially due to the new feature of allowing multi-character switches (complete words such as IMAGE as an alternate to the /I option for PIP).

Here's hoping either a patch can be found for CCL to correct the above problem or at least maybe the next release can have this straightened out.

Regards,



Ken Maxfield
11707 Macrinus
Florissant, MO 63033

Mr. Robert Hassinger
Coordinator - 12 Bit SIG
c/o: Liberty Mutual Research Center
71 Franklin Road
Herkinton, MA 01748

Dear Bob:

This letter is merely food for thought, to add to several other WISHES that have been mentioned in recent newsletters. It concerns the many versions and types of Operating Systems that are currently in use, available, etc. for the PDP-8 family of computers.

There are OS/8, OS/78, WPS-8, COS-310, ETOS, etc. Each of these operating systems have many good features (if they didn't, you couldn't even give them away). The main problem with these and other operating systems available for the PDP-8 computers is that (with a few notable exceptions) the programs and facilities they contain are not directly compatible with each other. WPS-8 cannot directly read OS/8 files or write them either. They have different MONITORS, file structures, handlers, etc. Even DEC (or maybe especially DEC) is adding to the proliferation by causing OS/78 to gradually drift away from OS/8 (for instance the former includes COMMERCIAL BASIC but does not include BUILD). You can solve the CURRENT differences in OS/8 and OS/78 by buying both and combining them for a version called OS/8-78 (I guess), but this may only be a temporary possibility (who knows?).

As an example, I currently have access to WPS-8 on a temporary basis. On another system which has OS/8, I have a manual for a program that is approximately 470 OS/8 blocks long. I would like to update-upgrade the manual using WPS-8 while I can, but unless I am willing to type the whole thing over again, there is no way -- perversity prevails !!

Naturally what I would like to see would probably take a superhuman initial effort, but I think the long run payoff would be unbeatable. I would like the several systems to be combined under a single Operating System - possibly as extension kits or whatever, but at least all of these facilities would be runnable under one OS. Actually the effort required may not be as "superhuman" as it seems, if a good deal of "gray matter" is used prior to starting on such a project. For instance, over the last two years I have been developing a standardized OS/8 program nucleus which can be used to write machine language programs in a style similar to high level languages. I have been able to use this nucleus to "resurrect" old "moth-eaten", stand-alone programs and convert them to full blown, device independent OS/8 programs in as little as 2 to 4 hours.

Mr. Robert Hassinger

Taking another look at the problem of no compatibility between operating systems programs, it appears that Mr. McIntyre's version of RUNOFF for the PDP-8 (which is now available thru DECUS folks!) was an appropriate development because there is no similar means to accomplish its function under OS/8 - OS/78. However, if the WPS-8 software would run under the same operating system, would the need for a RUNOFF have been nearly as great? I don't think so. What if the same effort had gone into developing some "unnamed" program that is not "available" through some other operating system? Where would we be today? FARTHER AHEAD! By the way, I don't intend this to be a "cheap shot" at RUNOFF - I think its great. My contention is that we are all forced to continue "re-inventing" the wheel because of the diversity (perversity?) of various development groups and operating systems and "profit centers" and etc, etc, etc. Remember - the problems at the TOWER OF BABEL were caused because the various systems, people, etc, couldn't communicate or work together. Is that where the PDP-8 operating systems are headed?

On a related subject, I was told just this last week by DEC that you can only purchase the WPS-8 software "WITH" a CPU. Does this mean that all of us who didn't think to purchase it "IN THE BEGINING" (or couldn't - because it didn't exist yet or because of limited "BUDGETS") will have to purchase another computer system if they want WPS-8?

Regards to all,

Ken Maxfield

Ken Maxfield
11707 Macrinus
Florissant, MO 63033

Note from Bill Haygood
On MULTOS/8 Multi-user OS/8

I appreciate the very generous response thus far received on MULTOS/8. Many have inquired as to support for other peripheral equipment and/or swapping disks other than the RK05. I would be glad to consider any requests that are made, but implementation of many of the requests would have to be on a time and materials basis. RLO1 and KL8A support is anticipated to be incorporated by 15 January 1978. Since the publication in the last 12-Bit Newsletter, I have added support to run multi-user OS/8 on SCIDATA supplied systems with Data Printer line printers and Pertec disk drives.

I am also an authorized distributor of DEWAR software for the PDP-8. Among these truly fine products is a macroassembler called PAGE8 which solves the PDP-8 paging problem--the PDP-8 is addressed as a 32K machine. Although PAGE8 source code looks amazingly like PAL8 code, PAGE8 offers many powerful features not found in other PDP-8 assemblers. An editor is available which features powerful string editing

functions and is extremely fast (it will close out a 250 block file from the first page in less than 3 seconds--even under timesharing, it takes only 4 seconds!). ACID is a document generator program which requires almost no markup commands and can even produce double column pages with automatic justification and auto-hyphenation. In fact, this 12-Bit Newsletter entry was produced by ACID with only two markup commands and no other information in the source file (not even hyphenation info) except the text. HIBOL is a superset of OS/8 BASIC and allows symbolic references in place of or in addition to line numbers. If a line is not referenced elsewhere in the program, it need not have a symbolic tag or a statement number.

I can also supply 8K or 16K semi-conductor RAMs on a single quad module. The current drain is less than other RAMs on the market. The price, too, is very attractive.

Interested persons are welcome to contact me on any of the above items at ComServ Enterprises, 7822 Oakledge Road, Salt Lake City, UT 84121, USA or calling me at 801-942-2300.

Parametrized batch files in OS/3

by W.L. van der Poel

University of Technology, DELFT, The Netherlands

The BATCH processor provided under OS/8 is of a rather simple minded kind. It can only execute a predetermined file of commands not allowing parametrization, let alone conditionals, indirect command strings, loops etc. Even very simple-minded systems for micro processors such as CP/M have the parametrization. We shall not strive for all facilities such as are offered to shell commands under UNIX on the PDP 11, but we shall show in this short article how parametrization of batch files is possible under OS/8.

We shall give two examples of essentially the same idea:

- 1) A general parametrizer for any given batch file with formal parameters.
- 2) A simpler process for a specific batch file.

An example of case 1) could be the batch file:

```
$JOB
.R FORT
*?1/G           ?1 stands for first formal parameter to be replaced
.TYPE ?1
$END
```

If now this batch file is available under the file name FORTYP.BM

it can be parametrized and executed with

```
.MUNG GEN,FORTYP,EXAMPLE           where EXAMPLE is the specific
program name of a FORTRAN program to be run and GEN.TE is the MUNG macro
doing the general parametrizing. In fact the following batch file is
actually executed:
```

```
$JOB
.R FORT
*EXAMPLE/G
.TYPE EXAMPLE
$END
```

The second case deals with MUNG files for running a specific batch file. In that case the call can be simplified to

```
.MUNG FORTRAN,EXAMPLE
```

Before being able to run these parametrizers one has to initialize the environment once only with .SUBMIT PREPARE where the file PREPARE.BI is reading as follows:

HK GY J 2SEX\$.,ZXA	Skipped as batch file. Place harmless
HK GA EWDUMMY.RA\$ EX	file DUMMY.RA
*200	dummy directive
\$JOB	Real beginning of batch file
.COPY RALF.SV < PAL8.SV	Copy PAL temporarily as RALF
.MUNG PREPAR.BI	Same file serves as MUNG file!!
.EX DUMMY.RA/Q	Execute DUMMY as PAL file. Will serve
.DEL DUMMY.*	to remember EXECUTE command
.COPY RALF.SV < BATCH.SV	Copy BATCH under the name of RALF
\$END	Could be another program, not used.

Once set up the EXECUTE DUMMY.RA/Q will be kept in the remembrance area of CCL.

The MUNG file GEN.TE will now be described.

```
J :S,$ "E HXA ON$" -D B,.XA B,.K
J :S,$ "E HX1 ON$" -D B,.X1 B,.K
HX2 !N! HK GY J 2SMZ$ B,.K HXZ MZ
HK GZ J 2S?A$ -2D GA J 2SMB$ B,.K HXB MB
HK ER?A.BM$Y
J < S?1$ -2D G1 >
J < S?2$ -2D G2 >
EWDUMMY.RA$ EG
```

When entering MUNG the parameters (FORTYP,EXAMPLE in the example) are in the text buffer. They are searched on the appearance of comma, the comma is deleted with -D and the first parameter (zeroth paramter) is stored in Q-reg A as the name of the batch file to be executed. If there is no comma, the condition "E HXA does it and goes on immediately to label N. In the same way the parameters 1 and 2 are separated and placed in Q-reg 1 and 2. If necessary, more parameters could have been separated in the same way. Now after label N the whloe macro is fetched again with HK GY, everything before and including the second MZ is chopped off and the remainder is placed in Q-reg Z and re-executed with HXZ MZ. In the next line the remainder is scanned for the formal name of the batch file to be executed (?A) and the actual name is substituted. Again the commands just done are chopped off and the remainder is placed in Q-reg B and re-executed with MB. Now the parametrized bacht file is read in with ERFORTYP.BM . With the repeat clauses < > all occurrences of ?1 are replaced by the actual first parameter and the same for the second parameter. This is written under file name DUMMY.RA . At last EG will close the batch file to be executed and the EXECUTE command will do just that by chaning to RALF.SV, which actually is BATCH.SV and is recognized by the .RA extension on the file DUMMY.

The specialized MUNG files for a particular process are much simpler to compose. We shall give here the example of a file FORTRAN.TE doing the same as the first example. Most of the dollar symbols will stand for altmode as usual in TECO macros except the \$ in \$JOB and \$END in the model text of the batch file.

HXA HK GY J 2SMZ\$ B,.K HXZ MZ	Store the only parameter, chop and reexecute.
HK GY J 2SEG\$.,ZXC	Store model of batch file in C
HK GC J < S?1\$ -2D GA >	Replace all occurrences of ?1 by first parameter, store as file DUMMY.RA and execute with EG
EWDUMMY.RA\$ EG	
\$JOB	
.R FORT	
*?1/G	Fortran compile and go
.TYPE ?1	Also type file
.R BOOT/DK	Reboot to end batch. See further
\$JOB	If FORTRAN was not on disk, the
.COPY SYS: < DTA1:FORT.SV,SABR.SV,LOADER.SV,LIB8.RL	
.R FORT	previous job fell through and the
*?1/G	FORTAN set of programs is reloaded
.TYPE ?1	from DTA1: (for example)
\$END	The reboot prevented the second job to be executed when the first one succeeded.

In a certain way we have made a conditional batch process here!

Numerous variations can be made to the technique shown here. In our environment it serves as a convenient method to set up all programs necessary to run different languages by inexperienced students, who only have to know how to .MAKE PROGRAM and then run it with .MUNG ALGOL,PROGRAM or .MUNG FORTRAN,PROGRAM or .MUNG LISP,PROGRAM as the case may be.

Unfortunately I have not been able to do away with the PREPARE program. If one is willing to change CCL then e.g. the extension .BM could be used to recognize these batch macros

For those not so well acquainted with TECO, let me remind you that these texts, containing altmode symbols, can only be input with the at construct e.g.:

Ⓐ I \ any text containing single altmodes but no backslash \

PUTEAUX, le 18th October 1978

PAGE 43

BP/YN.

Mr. Robert HASSINGER
Coordinator - 12 Bit SIG
LIBERTY MUTUAL RESEARCH CENTER
71 Frankland Road
HOPKINTON, MA 01748
U.S.A.

MOVING DATA BETWEEN 8'S AND 11'S THROUGH DISKETTES.

Dear Mr. Hassinger,

From the note of JIM VAN ZEE in the September newsletter (n° 30), it seems that CARL APPELOF and JIM succeeded in transforming with a PDP 11 a source file on diskette in such a way that it could be read on a PDP/8, using the STEWART DEWAR handler.

Since we have here some PDP 8 and 11, I did very recently the same but without any action on the PDP 11 side, except picking up the address and the place of the file. My PDP 8 program takes the 11 diskette as it is, translate it into one or several OS/8 file(s). I started investigating that way as soon as I received from JIM, when he passed in Paris in September, the STEWART DEVAR OS/8 handler mentioned in the newsletter 29.

As this handler uses the 8-bit mode transfer of the RX8, which means the full information capability of a diskette (667 blocks of 256 12-bit words = 77 tracks x 26 sectors x 128 bytes), then the conversion became theoretically possible between 12-bit and 16-bit words through RX8 and RX11 interfaces.

Having found that the bytes-packing was compatible with the RT/11 handler, and deciphered the relation between corresponding sectors (not so simple), I have written a small UWFOCAL program which asks the starting block number and the number of blocks of the diskette file, such as they are given by RT/11 PIP, then asks the device and name of the OS/8 file to be created, and finally operates the transfer sector after sector.

It could easily be improved, but, as it is, it does the job. I join a copy of the text, but I also add a paper about the method, so that anybody can write a program in his usual language, the only condition being that this language be able to access absolute blocks.

.../

The RT/11 DEC handler reveals only 494 blocks of 256 16-bit words, i.e. 76 tracks × 26 sectors × 128 bytes, with the same lose of one track as the OS/8 usual DEC handlers. It means that if you can translate with no limitation from 11 to 8 (with the STEWART DEWAR handler), it is better, when translating from 8 to 11 to drop starting blocks less than 11 (octal).

An RT/11 program can easily be written to do the symmetrical operation on 11 side, taking the OS/8 diskette as it is (provided that it has been prepared with the STEWART DEWAR handler).

The method is also given in my paper.

Yours sincerely,

A handwritten signature in black ink, consisting of several overlapping, stylized lines that form a cursive or shorthand representation of the name Bernard Perrette.

Bernard PERRETTE

METHOD TO CONVERT RT/11 SECTORS INTO OS/8 SECTORS ON DISKETTE

(BERNARD FERRETTE, BANQUE DE FRANCE, PARIS, OCTOBER 1978)

PREREQUISITES:

- 1-USING THE STEWART DEWAR 8 BIT-MODE OS/8 HANDLER,
FOR THE DRIVE RECEIVING THE 11' DISKETTE
- 2-KNOWING THE RT/11 ABSOLUTE STARTING BLOCK NUMBER OF THE FILE
- 3-KNOWING THE NUMBER OF RT/11 BLOCKS
- 4-OPENING SOMEWHERE AN OS/8 OUTPUT FILE

METHOD:

- 1-COMPUTE THE SECTOR NUMBER NS CORRESPONDING TO THE BLOCK NUMBER BD
- 2-COMPUTE THE TRACK NUMBER RP=INTEGER PART OF NS DIVIDED BY 26
- 3-COMPUTE THE SECTOR RANK RS IN THE TRACK (0 TO 25):
 $RS=NS-(26 \times RP)$
- 4-COMPUTE THE LAG D, IN TERMS OF 8-SECTORS, OF THE FIRST 11-SECTOR
IN THIS TRACK : $D=3 \times RP - (13 \times \text{INTEGER PART OF } (3 \times RP / 13))$
- 5-COMPUTE THE LAG AS OF THE SECTOR WITH RANK RS :
 $AS=RS+6.5 \times (1+\text{SIGN OF } (RS-12+D)-\text{SIGN OF } (RS-12)+\text{SIGN OF } (RS-25+D))$
BUT $AS=RS$ IF $D=0$
- 6-COMPUTE THE OS/8 BLOCK NUMBER WHICH COMPRISES THE 11-SECTOR
 $B8=\text{INTEGER PART OF } ((26+(RP \times 26)+AS)/3)$
- 7-COMPUTE THE RELATIVE SECTOR NUMBER TB (0,1,2) INSIDE THE BLOCK NUMBER
 $TB=26+(RP \times 26)+AS-(3 \times B8)$
- 8-INPUT THE B8 BLOCK AND COPY JUST THE TB SECTOR
- 9-ITERATE OR CLOSE

METHOD TO CONVERT OS/8 SECTORS INTO RT/11 SECTORS

SIMILAR, BUT:

- REPLACE NS BY NS-26 IN COMPUTING RP AND RS
- CHANGE THE STEP 5 BY :
- $AS=RS-D+6.5 \times (1-\text{SIGN OF } (RS-12+D)+\text{SIGN OF } (RS-12)-\text{SIGN OF } (RS-25+D))$
- CHANGE THE STEP 6 BY :
- $B11=((RP \times 26)+AS)/4$
- CHANGE THE STEP 7 BY :
- $(RP \times 26)+AS-(4 \times B11)$

C U/W-FOCAL: 16K-V4 10/16/78

01.01 C/IMMIGR, POUR CONVERTIR DISKETTE 11 EN DISKETTE 8
 01.02 C/BERNARD PERRETTE, BANQUE DE FRANCE, PARIS, OCTOBRE 1978

02.10 T !"CONVERSION 11 VERS 8"
 02.12 T !"METTEZ LA DISQUETTE 11 SUR L'UNITE 1"
 02.14 A !! "ADRESSE DU BLOC DEPART EN PIP RT/11, OPTION (WEL), ="BD
 02.15 S BD=F(14,BD)
 02.16 A !"NOMBRE DE BLOCS A TRANSFERER ="NB
 02.17 T Z3,"=" ,FITR(4*NB/3)+1" BLOCS OS/8"
 02.20 T !! "NUMERO DE L'UNITE OU SERA CREE LE FICHER OS/8"
 02.21 T !" (S POUR SYS) ="
 02.22 S C(7)=FIN()
 02.24 T !! "NOM DU FICHER OS/8 A CREER (A TERMINER PAR CARR.RETURN)"
 02.26 T !" (L'EXTENSION ,DA SERA AUTOMATIQUEMENT AJOUTEE)" ;D 15
 02.30 D A;I (211-C(7)),2,34
 02.32 D O R(-C(7));(C(1),C(2),C(3),C(4),C(5),C(6)),E;G 3,05
 02.34 D O SYS:(C(1),C(2),C(3),C(4),C(5),C(6)),E

03.05 S NB=4*NB
 03.10 S NS=4*BD
 03.20 S RP=FITR(NS/26)
 03.30 S RS=NS-26*RP
 03.40 S D=3*RP-13*FITR(3*RP/13)
 03.50 S AS=RS+D-6,5*FSGN(D)*(1+FSGN(RS-12,1+D))-FSGN(RS-12,1)+FSGN(RS-25,1+D))
 03.60 S BB=FITR((26+RP*26+AS)/3)
 03.70 S TB=26+26*RP+AS-3*BB
 03.80 J (TB+1)6,1,6,2,6,3
 03.90 Y NS,-NB;I (NB-.1)5,1;G 3,2

05.10 D C;O I,E;Q

06.10 D 6,7,6,9
 06.20 D 6,7,6,8,6,9
 06.30 D 6,7,6,8,6,8,6,9
 06.70 O I R1;<BB>
 06.80 F I=1,128,1;X FIN()
 06.90 F I=1,128,1;X FOUT(FIN())

14.05 Z N;I (#-1000)14,1;S Z=FITR(#/1000),#=#-Z*1000,N=N+Z*512
 14.10 I (#-100)14,2;S Z=FITR(#/100),#=#-Z*100,N=N+Z*64
 14.20 I (#-10)14,3;S Z=FITR(#/10),#=#-Z*10,N=N+Z*8
 14.30 S #=#+Z

15.10 F I=6,-1,1;S C(I)=-192
 15.20 S C=FIN();I (C-255),15,3;I (C-141),15,4;S I=I+1,C(I)=-C;G 15,2
 15.30 S C(I)=-192,I=I-FSGN(I);I (C)15,2;T '/' ;G 15,2
 15.40 R

PUTEAUX, le 19th October 1978

PAGE 47

BP/YN.

Mr. Robert HASSINGER
Coordinator - 12 Bit SIG
LIBERTY MUTUAL RESEARCH CENTER
71 Frankland Road
HOPKINTON, MA 01748
U.S.A.

MOVING DATA BETWEEN 8' AND 11' DISKETTES

Dear Mr. Hassinger,

In addition to my letter of October 18th, 2 new points :

1. Having just read EARL T. ELLIS' letter in newsletter 30, page 26, I start doubting whether JIM VAN ZEE really passed me the STEWART DEWAR HANDLER ? It could rather be the handler of Dr. LYNCH of XEROX, since EARL T. ELLIS gratifics it with 666 blocks, and only 650 blocks for the STEWART DEWAR'S. Mine definitely has 667 blocks (directory + 660 free blocks). Please ask JIM ...
2. As already said, the first track (= 26 sectors = 8 full OS/8 blocks = OS/8 blocks 0 to 7 included) remains unknown from the RT/11 handler.

Since the OS/8 directory uses blocks 0 to 6 included, then you can have simultaneously on a diskette :

- 1) an OS/8 directory
+ 1 OS/8 file 1 block long
+ 1 OS/8 "empty" 20_{10} blocks long
+ anything along 639 OS/8 blocks
- 2) an RT/11 directory
+ the same thing in a different sequence, along 460_{10} RT/11 blocks.

If you baptize with some name the OS/8 "empty", then you can, according to the place of your latest work, (PDP 8 or PDP 11) have RT/11 files or OS/8 files one another convertible without alteration of the "other" directory.

Yours sincerely,


Bernard PERRETTE

PS. - Copy of both letters sent to LARS PALMER.

Dear Bob:

I've tried lots of mechanisms to bypass the typewriter, but this one is the most immediate one to date: text entry via UWV's SCROLL and output via UW FOCAL thru Houston plotter!! Not terribly fast, but nice graphics and you get upper/lower case. Perhaps Jim Van Zee will offer multiple fonts like 12-point fun Gothic and italics. Oh yes- underlining and super/subscripts!

Re Mr. Thompson's comments in newsletter #29, p. 16, I recommend the technique described (and Mr. Ellis' elaborations) highly. And it IS faster. In our specific one-of-a-kind application, the FPP was there, an artifact of an earlier programmer's desire to run the ancestor of FORTRAN IV, and the decision was made to farm out some of the computation load to it. Our PDP-12 is now displaying 4 512-point running averages and calculating/converting five scoring parameters while digitizing eight ADC channels and replicating six of them on DACs along with various miscellaneous tasks such as real-time clock service, running twelve alarm/signal relays, scanning six sense lines, and running the line printer. The core restrictions which led to the use of FPP fixed-point also preclude doling out 2K for an average display buffer, so the running average is calculated 64 points at a time, and it's nice that the FPP-12 can at least absorb that load. Double-buffering would help; I'm trying to free up a second buffer so that CPU-CPU interaction is improved. So it's a really special situation and one that likely could have used Mr. Thompson's technique up to the point that a running average display was added. Since few FPPs seem to be running in fixed-point, I thought I'd note some techniques for using it in calculations, not just moving stuff around in RALF programs. My conclusion is that it is interesting, but it would've been cheaper to buy another 16K of memory for floating-point storage...

In newsletter #28, p. 33, I asked if anyone knew why an appropriate programmed delay would not prevent the loss of portions of PDP-12 characters when data breaks are in progress. No response. Examining the prints and manuals reveals that the PDP-12 display control contains a number of switch options, some of which are apparently devoted to settling time. The documentation isn't terribly clear. Before re-inventing the wheel, is there anyone out there with a few pointers regarding this situation? Incidentally, I've rigged up a remote display using the PDP-12 EXTENSION SCOPE outputs, a Tektronix 5103N 'scope, and three LM318 op-amps (741s and even a 4136 were too slow). Again, folks out there must have faced the same problem- please drop a line to Mr. Hassinger with a quick schematic or description because when the time came, I couldn't find the prior work in print anywhere. This is what the newsletter is for- to exchange information not available in the manuals and write-ups.

(plotted from an OS/8 file by UW FOCAL on Houston Omnigraphics,
model 6650)
(text entered via SCROLL editor)



File Référence

October 10, 1978

Mr. R. Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, Ma. 01748
U.S.A.

Dear Bob:

A brief note re three 'problems' in the Aug-Sept. Software News:

- (1) p. 9, FOTP fix: Wrong address for the version No.
This should be 15036, not 17236.

Hence: 15036/7101 7102

- (2) p.33, FUTIL fix: (This combines the two from Jim Crapuchettes
and myself)
Omitted: 12151/XXXX 1175

- (3) p.37, FRTS modification for new 'standard' 2-page system handler:

17534/3761 1570 not 1750

Last address is wrong: should be 17576

Hence: 17576/6222 6220

not 17566/0212 6220 !!

I phoned Jim Mechtel about these and he was suitably grateful. He says they're having problems with a new method of preparing Software News. He also told me they have a modified version of BASIC to deal with the 2-page system handler problem - the present one is a mess.

Yours sincerely,

I. M. Templeton

USCG Research and Development Center
Avery Point
Groton, CT 06340

November 2, 1978

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

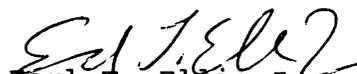
Dear Bob,

I have recently found some patches to PAL8 which may interest the SIG. These will work for V3C, V3D, and ETOSV5, (PAL8 V9 and V10) I have not checked other versions. Patch 1 is to vary the amount of '200' code that is output to non-file structured devices from the 64 now being output. The key is in an area of initialization code. At location 02176 there is a 7710 (-70[8]). By making this more negative, more '200' code will appear, less negative and less '200' code. This literal is only called from 02003, so any value can be used.

The second patch has to do with the number of columns PAL8 outputs. The default is 4, and '/7' will allow 7 columns. My patch inverts the '/7' switch so that default (4) is only used with '/7', all other times PAL8 uses the columns coded by this patch. First change 05656/7650 7640 which will change a SNA CLA into a SZA CLA to invert the switch. Next insert into location 05765 Minus the number of columns (octal). I wanted 5, so I put in 7773. Into 05763 put the 'Offset to First Symbol on Next Page' (Octal). PAL8 puts 55[base 10] symbols in a column, so the 'Offset' is equal to 55 times (number of columns-1) converted to octal. I wanted 5 so I had 55 times 4 = 220 = 330(octal), and put 0330 into 05763. 5 Columns is very nice if your Lineprinter is 80 characters wide. The 5th column will end just under the space after PAGE.

I am looking for an Inventory Management Program to run with OS/8. I assume it will be a BASIC or FORTRAN program. If anyone has such a program, I would like to swap for it, please write. I am also looking for a OS/8 handler for the TC58 Magtape (OS/8 V2 or earlier I think).

Thank you,


Earl T. Ellis Jr. Rm 143

(203) 445-8501 ext 296

USCG Research and Development Center
Avery Point
Groton, CT 06340

November 2, 1978

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

Dear Bob,

I have recently found some patches to PAL8 which may interest the SIG. These will work for V3C, V3D, and ETOSV5, (PAL8 V9 and V10) I have not checked other versions. Patch 1 is to vary the amount of '200' code that is output to non-file structured devices from the 64 now being output. The key is in an area of initialization code. At location 02176 there is a 7710 (-70[8]). By making this more negative, more '200' code will appear, less negative and less '200' code. This litteral is only called from 02003, so any value can be used.

The second patch has to do with the number of columns PAL8 outputs. The default is 4, and '/7' will allow 7 columns. My patch inverts the '/7' switch so that default (4) is only used with '/7', all other times PAL8 uses the columns coded by this patch. First change 05656/7650 7640 which will change a SNA CLA into a SZA CLA to invert the switch. Next insert into location 05765 Minus the number of columns (octal). I wanted 5, so I put in 7773. Into 05763 put the 'Offset to First Symbol on Next Page' (Octal). PAL8 puts 55[base 10] symbols in a column, so the 'Offset' is equal to 55 times (number of columns-one) converted to octal. I wanted 5 so I had 55 times 4 = 220 = 330(octal), and put 0330 into 05763. 5 Columns is very nice if your Lineprinter is 80 characters wide. The 5th column will end just under the space after PAGE.

I am looking for an Inventory Management Program to run with OS/8. I assume it will be a BASIC or FORTRAN program. If anyone has such a program, I would like to swap for it, please write. I am also looking for a OS/8 handler for the TC58 Magtape (OS/8 V2 or earlier I think).

Thank you,

Earl T. Ellis Jr. Rm 143

(203) 445-8501 ext 296

October 12, 1978

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

Gentlemen:

We have obtained such useful and encouraging information from the 12-bit SIG Newsletter, that we thought we might contribute a discovery of our own:

The PDP-8A executes a 7014 (RAL RAR) so as to join load the PC into the AC. We believe this instruction can therefore greatly facilitate the writing of relocatable programs.

Of course, DEC has refused intelligent comment on several occasions, so, I'm not sure this instruction will always exist.

Good Luck and Viva PDP-8!

Yours truly,



David Reynolds

THE JERSEY JOURNAL
JERSEY CITY, N. J. 07306

SAM NEWHOUSE

October 19, 1978

Mr. Robert Hassinger
Coordinator - 12 Bit Sig
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA. 01748

Dear Sir:

I have a PDP 8-I running OS-8. I urgently desire to hook it to an auto-answer modem so that it can be called up on the phone. I have been unable to locate an auto-answer modem with passive current-loop interface, which would be necessary because the computer is an active current loop device.

Help! Cost is no object. The only proviso is I must retain ability to replace auto-answer modem with TTY when needed (for loading paper tapes).

Sincerely,

SN/kf

Sam Newhouse
The Jersey Journal
30 Journal Square
Jersey City, N. J.
07306



4300 Alton Road, Miami Beach, Florida 33140
Tel. 674-2121
674-2610

October 24, 1978

PAGE 54

Mr. Robert Hassinsner,
Coordinator - 12 Bit SIG
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, Ma. 01748

Dear Bob:

I wish to thank you for the outstanding job you are doing as coordinator of the 12 BIT SIG. We are indeed fortunate in having your strong "voice" speak out for us in matters concerning the continued support of DEC 12 Bit computer.

Presently we have two LINC-8 and two PDP-12 computers. The two LINC-8's are being replaced by PDP-11's. Therefore, I would be most appreciative if you would announce the availability of these machines in the next Newsletter. Both of the LINC-8's are currently under DEC service contract and are in fine condition. Both machines are identical in hardware with the exception that one has 8-K memory. The machines are available immediately. Anyone interested should contact me at the above telephone or address.

Thank you for your help.

Sincerely,

A handwritten signature in cursive script that reads "Neal D. Atkins".

Neal D. Atkins
Systems Analyst

NDA/b

FOUNDATION FOR BLOOD RESEARCH

P. O. BOX 426

SCARBOROUGH, MAINE 04074

PAGE 55

October 10, 1978

Robert Hassinger, Coordinator - 12 Bit SIG
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748

Dear Mr. Hassinger:

We wish to announce that we will be selling a PDP-12 system within the next six to twelve months. This is a fully configured system, including two RK8 disk drives, FPP, and line printer. I have enclosed a more detailed description of the configuration. The system has been under continuous DEC service since its purchase in 1970 and is in perfect running condition.

Sincerely,

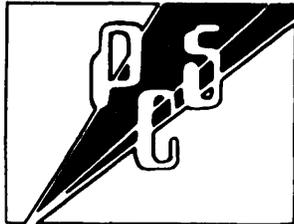


Dwight Smith
Systems Analyst

ph: (207) 883-4362

Digital Equipment Corp. PDP-12

<u>quan.</u>	<u>item</u>
1	PDP-12A cpu w/12k memory
1	ASR33 teletype terminal
2	RK8 disk drives (800kw/drive)
1	FPP12-AB floating-point processor
1	LP08-FA line printer (80 col, 64 char set, 356 l.p.m)
1	CM8I optical mark card reader
1	KW12 real time clock
9	RK01 disk cartridges (8 sector, 1100 bpi)



PROCESS CONTROL SYSTEMS, INC.

18130 S. Thornapple Lane

New Berlin, Wisconsin 53151

PAGE 56

(414) 782-3945

HARDWARE CONSULTATION

October 12, 1978

SOFTWARE DESIGN

PROCESS CONTROL SYSTEM DESIGN

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

Dear Bob:

Here is a patch to CREF14, Version 2C, which upgrades it to Version 2D. The patch is to solve a slight problem with shift register coils not being found in some cases.

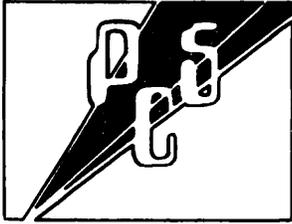
Patch to update CREF14, V2C to V2D.
This corrects a problem with contacts referenced only within shift registers.

```
.GET SYS CREF14
.OB
2674/5243      5712
2712/XXXX      2643
4211/3133      4306
4306/XXXX      0
4307/XXXX      3133;1313;3714;5706;2624;2712
4254/3133      4315
4316/XXXX      3133;1322;3714;5715
0002/6203      6204
^C
.SA SYS CREF14
```

Yours truly,


Michael E. Mazzone
President

MEM:blm



PROCESS CONTROL SYSTEMS, INC.

18130 S. Thornapple Lane

• New Berlin, Wisconsin 53151

• (414) 782-3945

HARDWARE CONSULTATION

October 12, 1978

SOFTWARE DESIGN

PROCESS CONTROL SYSTEM DESIGN

Dr. Eugene J. M. Lynch
 Xerox Corporation
 Joseph C. Wilson Center for Technology
 Rochester, New York 14644

Dear Dr. Lynch:

Thank you very much for your letter of 12-AUG-78 and the listings. About a month ago I borrowed a floppy disk drive and tried out the system and non-system handlers and they worked just fine.

I found that the non-system handler I discovered and your handlers were not compatible: they interleave sectors differently. Your handlers are definitely superior. I have abandoned use of the DAD handler and have advised my customers of the existence of your handler.

Your handler ran with OS/73 on a PDP-8/E using a DAD floppy disk drive. For the system handler to work on a VT-78 it would have to be bootable using the code in the built in bootstrap in the VT-78. Since your handler puts block 0 on track 0 and bootstrap expects to find block 0 on track 1, a system won't boot directly on a VT-78. The technique you described for booting on a PDP-8/A would have to be used.

Yours truly,

Michael E. Mazzoni
 Michael E. Mazzoni
 President

copy: Hassinger

~ The '8' Men ~

108 William St.

Wallingford, CT

06492

Dear Bob,

We've written to tell you about "The '8' Men," a group of PDP8 users who have pooled their resources to make available to other 8 users software that large corporations wouldn't normally market. Presently, we're the only members. We hope that expansion will follow as we get organized.

We currently access the Choate Computer Center: a PDP8/M, 24K of core, 4 TTYS, 3 CRTs, 2 phone lines, and 2 TD8E drives. We run an augmented version of ETOS, part of which is described below. As further modifications of ETOS are debugged, or any additional original software is perfected, The '8' Men will market it. A new multi-field, expanded BASIC is in the works even now. Some of its features include TAGs instead of line numbers, expanded editing, and some new and interesting additions to the command set.

Our first release: BASIC OS or The BASIC Operating System. The new system consists of three parts. The first part is a modified version of the LOGIN program. Instead of only ETOS accounts being legal in response to 'ACCOUNT?' you can now input an 'User Account. This new type of account is composed of two letters followed by six digits. Each new user account is stored and updated in the ETOS file 'USERACC.NTS' - hence 1024 accounts can be stored and used without the normal ETOS overhead.

Once logged into an user account, the user is booted to the second part of BASIC OS - a simplified version of OS/8. The keyboard monitor command structure is different, but any OS/8 compatible program will run under BASIC OS. The aim of the new monitor is to allow the neophyte easier access to the higher level languages and to protect the larger system from the accidents of the beginning user. To this end, CTRL/V has been disabled in the monitor. It may be re-enabled by running any OS/8 file which performs a SETSTAT with the CTRL/V bit on. Often the SCALE command set is more confusing than helpful to the beginner.

The final component of the system is an original program called 'UACCNT.SV.' It is the BASIC OS equivalent of the ETOS updating program ACCNT.SV. UACCNT allows the user to add, change, and delete user accounts as well as modify and list the contents of their charge words. These charge words are a running total of computer usage from \$00.00 to \$40.95. Other than their obvious use as one way to help your computer center to be self-supporting, the charge words can be used as an indicator of overall computer use. When each user logs out, he receives the amount of 'money' he used for terminal connect and CPU time as well as his running total. Each month the charge words could be listed and zeroed to provide a month by month record. Such an arrangement is ideal in an educational environment.

Since ETOS is a licensed product, we can sell this package only to those 8 users who have such a license. If the interested user wants BASIC OS on his/her own disk, send the disk to us and we will build BASIC OS onto it for \$40.00 (an extra \$20.00 for the source on the disk). Paper tape binaries and the manuals can be ordered for \$20.00, manuals only for \$3.00, source listings for \$10.00, or machine readable source for \$30.00. These prices apply if you send a deck of your own, but if you want the products on deck and don't send one, include an extra \$15.00. Without your own disk, expect to add \$100.00 to the above prices if you request the products on disk or the system to be built on disk. We are just starting out and would be grateful for readers' feedback or suggestions on this product or any future ones they might like to see.

EDUCOMP-QUODATA plans to put a small write-up of this system in their ETOS newsletter. We hope to hear from many of you!

Charles L. Perkins

William D. Cattery

~ The '8' Men ~

Charles L. Perkins
William D. Cattery

August 23, 1978 A.D.

CLP/clp

DATE FIX FOR U/W-FOCAL, DECUS FOCAL8-301

The date routine in U/W-FOCAL (Version 1) was not designed to work beyond the end of 1977 since no provision had been made at the time it was written (1974) for extending the system date beyond that time. Consequently dates in 1978 are printed as '1970', and in general, dates will cycle through 70-77 every eight years. The following patch may be used to add the necessary offset so that the proper year is shown. Note that the offset will have to be changed every eight years, and in any case will not work after 1999.

```

.GE SYS UFOCAL          (see listing pp 16, 16-1)
.ODT
14510/ 7440 7000        NOP
14460/ 4317 0767        AND I (7666
14461/ 1767 1264        TAD BASEYR
14462/ 0366 4274        JMS PACK2
14463/ 4317 7410        SKP
14464/ 1061 0116        BASEYR,      116
↑C
.SA SYS UFOCAL          (Do it twice to put the program
.SA SYS UFOCAL          back in its original location.)

```

Note that the last location, 14464, will have to be changed every eight years. The table below gives the proper value:

```

1970-1977: 0106
1978-1985: 0116      (Shown above)
1986-1993: 0126
1994-1999: 0136

```

The version distributed from the library will always have the current base year patched. Other versions (currently V4D) are available from the author.

15 August 1978 JvZ

DECUS HAS MOVED!!

As of August 14, 1978, the DECUS International Headquarters and DECUS U.S. Chapter offices will be located at Digital Equipment Corporation in Marlboro. Our new address is:

DECUS
MR2-3/E55
One Iron Way
Marlboro, Massachusetts 01752

Marlboro is not on Centrex, a direct access telephone system, so all calls will come thru the switchboard at (617) 481-9511.

DECUS extensions are as follows:

Central Number	4100
Executive Director	4120
Admin/Finance	4122
Order Processing	4135
Accounting	4136
Membership	4167
Publications	4131
Library	4178
U.S. Chapter	4141



DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MR2-3/E55
MARLBORO, MASSACHUSETTS 01752

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

- Change of Address
- Delegate Replacement

DECUS Membership No.: _____

Name: _____

Company: _____

Address: _____

State/Country: _____

Zip/Postal Code: _____

Mail to: DECUS - ATT: Membership
One Iron Way, MR2-3
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here.
Include name of installation, company, university, etc.