



DECUS 12 BIT SPECIAL INTEREST GROUP
NEWSLETTER

January

Number 38

1980

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-February-80. Material requiring editing/re-typing should be in earlier. Ready-to-use material should use an area 7 inches (18 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.

Material submitted in machine readable form is particularly desirable because it can be edited and incorporated into the newsletter format more easily. Higher quality reproduction is also possible this way. Contact the editor (Bob Hassinger) for further details on acceptable media and formats if you plan to make a submission in machine readable form.

FORMATION OF WPS-8 WORKING GROUP

At the Fall Symposium in San Diego, we decided to form a WPS-8 Working Group due the great interest that has been expressed in the WPS-8 Word Processing products. Steven A. Taylor has agreed to head the new Working Group. Check Steve's report in this issue on WPS activities at the Symposium for more details.

© 1979, DECUS

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

SIG COMMITTEES AND WORKING GROUPS

Steering Committee:

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

Jim Crapuchettes
Menlo Computer Associates, Inc.
801 E. Charleston Rd. Suite F
Palo Alto, Calif. 94303
(415) 494-3170

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

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

Lawrence H. Eisenberg
17141 Nance Street
Encino, California 91316
(213) 788-0354

Special Steering Committee Advisors:

Stan Rabinowitz

Tom W. McIntyre

Micro-8 Working Group and US Symposia Committee Representative

Jonathan Lockwood - see above

RTS-8 Working Group
Lee Nichols - see above

WPS-8 Working Group
Steven A. Taylor
Kubernan
Cloverdale Executive Building
2110 Cloverdale Avenue
Winston-Salem, NC 27103
(919) 725-1915

COS-310 Working Group
Lawrence H. Eisenberg - see above

Symposium Software Exchange Committee

Send copies of software you wish to exchange at the next US Symposium to one of the following committee members for preparation:

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

James Coryell
863 France
Simi Valley, CA 93065
(805) 526-7478

NOTE FROM WOLFGANG LEBER

"As I promised you when I was in the U.S.A. I have prepared two short reports for the SIG newsletters. One is the report on the meeting of the German speaking 12-Bit-SIG in the Max-Planck-Institut fuer Hirnforschung in Frankfurt/Main.

"The other is a technical note for users who want to connect a DSD 210 or 440 with a PDP-12 or any other pre-OMNIBUS machine.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

"As yet I have not prepared a discription of our "EURO-12" called Eurocard 6100 system because I am not sure that this would be of interest. Here a few remarks concerning the general design :

- CPU - 6100 with MEDIC, Bus tranceiver, Clock generator and Single step control logic
- 8 KW - Memory stack static NMOS
- Control panel software so far for RX01 and RX02 Bootstraps (in preparation CP-Monitor like a LSI11 with support for the VT/78 CP IOT's).
- TTY - Interface full DEC and VT/78 compatible with modification for the use of Kansas City Cassette and also Modem support.
- Floppy disk Interface for DEC Data transmision procedure
- Lineprinter Interface parallel (Centronics mode)
- 16 Channel 12 Bit-ADC, LAB8E compatible

"The whole in a 19" card cage for building in a rack. The system configuration contains also a DSD 440, a SWT low cost graphic - terminal and the ANNADEX 1200 Chr. Matrixprinter."

Wolfgang's address is Max-Planck-Institut fuer Hirnforschung Neurobiologische Abt., D 6000 Frankfurt/Main 71, Deutschorden Str. 46, West Germany.

GERMAN 12-BIT SIG MEETING IN FRANKFURT

The 2nd annual meeting of the German speaking 12-Bit-SIG of DECUS Munich was held at the Max-Planck-Institute of Brain Research in Frankfurt. About 45 members coming from several countries of central and south-eastern Europe attended the meeting, a fact that proves the strong activities of the PDP8/12 users here.

The main topics discussed and papers presented were:

- Report of the DECUS Europe Symposium which was held at Monte-Carlo just 2 weeks ago
- Information on DIGITAL's PDP8 policy especially concerning the move of the PDP8 to the Traditional Product Line
- Several concepts of small dedicated 6100 microprocessor systems presented by several users were discussed.

During the afternoon a survey was given on the existing multi-user and time-sharing operating systems. Comparisons, especially concerning the possible throughput, achieved by the different hardware/software approaches found a high interest. Discussed and compared were MULTI-8, MULTOS-8, ETOS and a dedicated German multi-terminal system for commercial applications which is based on the OS/8 handler structure for file I/O.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

The highlight of the meeting was without any doubt a demonstration of 7 (seven) PDP8 family computers. Ranging from low-end 6100 micro-processor kits, MICRO-12 from HARRIS and INTERCEPT one-boarders, a new German Eurocard-format 6100-based micro-computers the 32 kW MOS-RAM, called 'EURO-12', running OS/8 on a DSD440 double-density floppy-disk, and going upward to two DECstations running WPS-8 and OS/8 resp., equipped with RX02s and a line-printer.

Moreover, towards the higher end, a 2-terminal PDP8A, running the MULTI-8 time-sharing system, using a RK05, RX02 and a line-printer, a system heavily loaded by 'Adventure'-players during the coffee breaks.

The copying of some neat software packages, containing compilers, utilities and games was organized.

Obviously, everybody agreed finally, it has been an interesting and successful meeting, which surely gave a lot of ideas and stimulating suggestions to the attendants. It was demonstrated again, that the PDP8 is not dead at all, but rather alive!

W. Leber, B. Kup

HELP - FILE CONVERSION PROGRAMS

With the growing interest in COS-310 and WPS-8, we are seeing a great increase in interest in programs to allow moving files between all the various 12 Bit operating system formats. These include OS/8 files, WPS-8 "documents", COS files and COS logical units. Specifics of the media involved make a difference also (i.e. COS DECTapes are compatible with OS/8 DECTapes although the COS files use 6 bit rather than 8 bit representations for characters) but the same is not true in the case of floppy disks (the 12 bit mode vs. 8 bit mode plus interleaving scheme questions). I want to collect as much information as possible for the Newsletter on this subject. Input from anyone who can help will be greatly appreciated. I hope we can have a working session at the Spring Symposium to follow up on this. We also need inputs and a Symposium working session to better understand all the "byte mode" floppy handlers that are available.

Send your inputs to me (Bob Hassinger - see page 1 for address) as soon as you can. The next Newsletter deadline and the Symposium are coming up very quickly.

OS/8 SOFTWARE UPDATE INFORMATION

At the Fall Symposium in San Diego, DEC held sessions on the various OS/8 related software. They had hand-outs with the ordering information and prices. Certain items in particular caught my eye and I have copied the information we were given here.

OS/8 V3D COMBINED KIT

This was described as a repackaging of the entire OS/8 set. That is: OS/8, Extensions, Fortran IV, and Device Extensions kits all in one. Note that with the Device extensions kit I think you get the 128k support changes and so you get more recent versions of some things like CCL than are in the regular OS/8 V3D kit. This may be good or bad but you should be able to mix and match from this kit and your OS/8 V3D kit.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

Media	Supported Binaries	Unsupported Binaries (expired warranty)	Update Binaries (in-warranty)	Sources
RK05	QF024-AE \$1000	QF024-HE \$500	QF024-WE \$300	QF024-EE \$2500
RL01	QF024-AQ \$1000	QF024-HQ \$500	QF024-WQ \$300	QF024-EQ \$2500
DECtape	QF024-AC \$1000*	QF024-HC \$500*	QF024-WC \$36*	
RX02	QF024-AX \$1000*	QF024-HX \$500*	QF024-WX \$24*	
RX01	QF024-AY \$1000*	QF024-HY \$500*	QF024-WY \$48*	

* - JANUARY 1980

RTS/8 V3.0 - INCLUDES MACREL/LINKER V2A

V3 of RTS/8 requires MACREL/LINKER so it is packaged with it in this case. This may be the best way to get MACREL/LINKER in some cases.

Media	Digital Supported (Customer Installed)	Customer Supported (Customer Installed)	Unsupported Update (expired warranty)	Supported Update (in-warranty)
DECtape	QF028-XC \$900	QF028-YC \$450	QF028-NC \$200	QF028-VC \$24
RK05	QF028-XE \$900	QF028-YE \$450	QF028-NE \$350	QF028-VE \$100
RL01	QF028-XQ \$900	QF028-YQ \$450	QF028-NQ \$400	QF028-VQ \$150
RX01	QF028-XY \$900	QF028-YY \$450	QF028-NY \$200	QF028-VY \$16

I cannot tell which, if any of these prices include documentation. I have extracted some of the documentation prices from the handout. Note that the OS/8 set is the new one containing five 8 1/2 by 11 manuals which replaces the OS/8 handbook and update(s). DEC noted that the OS/8 Handbook we all know and love(?) is now a collector's item! Also notice the FORTRAN IV Software Support Manual - many people have never heard of it. Although some of it may be out of date a little, there is still useful information in it that is hard to find otherwise. This is not all the documentation, the handout for the OS/8 Combined Kit Documentation Set lists 14 manuals in all for a complete set. That does not include RTS/8 or MACREL/LINKER. You might also like the OS/78 manual set which is cheaper and smaller for where the more limited set of functionality covered is enough.

RTS/8 V3 User's Manual	AA-0724D-TA	\$11.00
RTS/8 V3 Release Notes	AA-5158B-TA	\$3.50
RTS/8 V3 SYSGEN Manual	AA-D943A-TA	\$11.00
OS/8 MACREL/LINKER Release Notes	AA-5663B-TA	\$3.50
OS/8 MACREL/LINKER User's Guide	AA-5664B-TA	\$9.00
OS/8 Documentation Set	QF015-GZ	\$85.00
OS/8 Device Extensions User's Guide	AA-D319A-TA	\$9.00
OS/8 Device Extensions Release Notes	AA-H565A-TA	\$3.50
OS/8 FORTRAN IV Software Support Manual	AA-4532A-TA	\$11.00
OS/78 User's Manual	AA-5748B-TA	\$14.00
OS/78 User's Manual Update	AD-5748B-T1	\$3.00
OS/78 System Release Notes	AA-D927B-TA	\$3.50
OS/78 Command Summary	AV-5582B-TA	\$1.00

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980DEC TRADITIONAL PRODUCTS REFURBISHING

I received the following note from T. H. Nilsson of the University of Alberta:
"Encouraged by the note on page 13 of issue #36, (on TPL services and how they refurbished our PDP-8/I - RH) I checked with the Traditional Product Line about refurbishing my recalcitrant PDP-12B. They refused to have anything to do with it and told me to buy an LSI-11 MINC. I explained that funds were available only for the repair of existing equipment, but this made no difference to TPL. Obviously there is a "breakdown in communications" within DEC and our SIG. How can I learn what the real situation is for Traditional Products?"

I tried unsuccessfully to call Mr. Nilsson so I contacted Ray Stimpson, the Marketing Manager for TPL. I think we agreed that TPL needed to communicate a little better with customers in this kind of situation. Ray doubted that anyone in TPL would suggest the MINC as a replacement for the PDP-12 in a case like this. It might do some of the work but it has far less capabilities in many areas, it cannot run the existing PDP-12 programs, and there would be a difficult problem even trying to move existing data files to a MINC because in most cases there is no common, compatible media available on these machines.

It appears that the TPL "charter" has changed a bit since they worked on our 8/I. They now can only take title to equipment, refurbish it, and then resell it rather than taking a customer's machine, refurbishing it and returning it to him. Perhaps TPL's experience with our machine helped contribute to the change in policy. I always suspected that job was not at all profitable for them.

In any case Ray contacted Mr. Nilsson and discussed the problem. Ray also forwarded the following explanation to me:

"Per our telephone conversation of December 3, we discussed Mr. Nilsson's letter in the November 12-Bit SIG Newsletter regarding his request for Traditional Products maintenance support for his PDP-12.

"As you know, Traditional Products is chartered to acquire (take title), fully refurbish, and re-sell into the general marketplace, Digital systems, equipment and options that are in current demand. In addition, any refurbished equipment from Traditional Products is further covered by a Digital warranty and is authorized for Digital Field Service.

"In my conversation with Mr. Nilsson to determine the exact nature of his requirements, we concluded that transfer of title and full refurbishment was not the most equitable and financially feasible solution.

"We concluded that a Field Service refurbishment on site appeared the most prudent course of action."

I presume from this that Field Service has improved its capabilities in this area over the last couple of years. Before finally having TPL work on our machine, I spent many frustrating months going around and around with our local Field Service office. It seemed that they had no idea how to do what needed to be done, did not have the resources or know-how to do it and they seemed very disinterested in the whole problem. It was only as a last resort that I went to TPL. Either this was a local people problem, or Field Service has now gotten that act together, or we will find that they still can-not do the job. I will be interested to hear which one it is.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

CORRECTIONS FOR THE OS/78-V3 MANUAL

Jim van Zee submitted the following comments and observations to DEC as an SPR and forwarded a copy to me.

The new OS/78-V3 manual is an astounding accomplishment for which I am sure the entire user community is most grateful. It is not unexpected, however, that such a large undertaking might have accidentally allowed a few errors to slip through, and while enjoying the book I also noticed the following items which I thought to bring to your attention:

1) The discussion of the use of the ESCAPE key is incorrect: PAL8 does not use this key, but other common monitor commands (which were not mentioned) do, viz. LOAD and MAP.

2) The use of .BK as an extension appears in several examples throughout the manual, yet is omitted from the table of 'official' extensions on p. 2-12 and Appendix E. This should be corrected so that the new user will recognize that .BK means 'backup file'.

3) The current date is NOT appended to files which are renamed or copied for which we are all sincerely grateful! The discussion in the DATE command, however, erroneously indicates the contrary. This discussion also contains some very bad advice about listing directories with old dates and contains a silly mistake in one example. (If no date is entered when the system is started, no dates appear in the directory listing. Set the system date to an earlier year in this case.)

4) A caution in the DELETE command is wrong: FOTP will not assume *.* for a null specification when the /D switch is set. This is treated as a special case to avoid accidental disasters. Try .DEL SYS: just to prove this to yourselves. Also, it is the INPUT specification which determines what gets deleted, so the *.* in the output specification in the next-to-last example is unnecessary.

5) The BITMAP example on page 3-48 is confusing at best. The text says 'Press ESC to terminate the last line and execute the command' which makes it look like you will get a map of some 12 files whereas you will only get a map of the last because of the /R switch. This should be pointed out explicitly. Best if this example were broken up into two examples to illustrate the use of ESC and /R separately.

6) A minor error on p. 3-60: the addresses are now 6-digit, not 5, and the SAVE example which causes an error does so because there are two file name specifications as well as an overlap in memory locations. Also the error printed is not 'SAVE ERROR' but 'BAD ARGS' (even if you remove the erroneous second file name). The former message (in my experience) comes only when there is insufficient room on the device for the file. Also, I think it is important to point out that memory is still intact after such an error so that you can try again. This used to be pointed out in earlier documents, and has comforted me on numerous occasions.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

7) The message about ESC in a BATCH file (p. 3-73) certainly confused ME, since I have often used the '\$' construction to terminate a command which would be terminated by ESC from the keyboard, and it certainly did not cause an error. However, I eventually figured out that you meant a REAL escape, which does indeed cause an error - but the solution is not to just tell people that they can't use ESC in such a situation, but to tell them that they should use '\$' wherever they would normally use 'ESC'!

8) Finally, on p. 5-24 there are a couple of errors. The most glaring is the incorrect figure for the "Memory Ref. Instruction Format", but a more subtle one appears in paragraph 5.9.2.5 in which a vague reference seems to imply that the use of '\$'s in text strings, etc. has been permitted only for compatibility with older versions and is not recommended. (You surely do not mean that TAD ("\$ is not recommended!))

I am sure there are other such glitches, but none that caught my eye as readily as these. Thanks for a great manual!

Jim van Zee

DECUS PROGRAM LIBRARY NEWS

The following program was added to the library in December.

DECUS 8-915 - PRINTR - A Text and Graph Plotting System with Expanded Capabilities. Submitted by Graham F. Forsyth of Aeronautical Research Laboratories, Victoria, Australia. Written in PAL-8, runs in 8k, uses a Versatec D800A Printer/Plotter or equivalent.

"A system for using control character sequences to extend a printer's character set and to more efficiently handle graph data and drawings has been written as a machine language program for the PDP-8 computer with electrostatic plotter. Features added to text files include superscripts, two levels of subscripts, overstruck bars, dots, and combinations and Greek and mathematical symbols. Plot files have two modes using 8 or 12 bit data words and may be either continuous scan or byte addressed. Plot and text files may be mixed but not superimposed (at present)."

Price code: K25. Documentation on the magnetic media.

NOTE FROM JIM VAN ZEE REGARDING THE 'SET HANDLER' COMMAND

The SET program being distributed with OS/78-V3 (SET-V3A) has a very significant new feature which solves one of the major design flaws in OS/8, namely the lack of sufficient handler space. The purpose of this note is to explain the problem to those who have not yet encountered it and to then illustrate how the new 'SET HANDLER' command may be used to overcome this long-standing difficulty. Although intended for use only on OS/78 systems, this new version of SET can be easily patched to work on older machines. A list of such patches is included at the end of the article.

The Problem: The key to much of the flexibility of the OS/8 operating system lies in the ability to separate the I/O portion of a program from the rest of the logic. This is accomplished through the use of standardized subroutine calls which are passed to an appropriate 'handler' in the monitor system. Space has been reserved in the system for eight such handlers, not including the system handler which is always resident in

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

memory. Thus the maximum number of different devices which an OS/8 program can communicate with is 8 + the system device. To accommodate devices which have more than one logical 'unit', the system actually allows for 15 different device names, so a single floppy disk handler, for instance, can be used to access either Drive 0 or Drive 1, depending upon which entry point is used.

These limits may have seemed reasonable 10 years ago when most systems had only dual DECTapes and a TTY, but the increasing use of large disks has made these limitations far more critical because of another restriction in the system, namely that no device can have more than 4095(10) blocks. Thus to take advantage of one of the new 45MB 'mini-Winchester' drives which are currently the rage would require over 28 different device names! A secondary reason for the 'handler crunch' is that more than one handler may be required for a given device in order to accommodate different data formats. Consider, for example, the lowly TTY. In addition to the standard handler used for ASCII data, a special handler is required for reading or punching binary tapes, and yet a third handler is needed for creating legible characters on such tapes for identification purposes.

Thus it doesn't really take a very fancy system to exceed the handler limitations. For instance, a PDP-12 system which I use frequently has the following devices: a hard disk, dual LINCtapes, dual floppies, a CRT, TTY, serial-line port and a line printer. Except for the floppies and the line printer this is a rather typical PDP-12 which at first glance would appear to need only 7 device handlers. However LINCtapes come in two flavors, so a second handler is necessary for reading DIAL tapes, while the same drives can also be used for reading DECTapes, which requires still another handler. Floppy disks likewise come in (at least) two different formats (see issue #36, pp. 8-11), and as mentioned above it is nice to have 3 different handlers available for use with the TTY. Thus there is a need on this system for some 10 different handlers with about 20 different entry points, not including any 'non-device' handlers such as BAT or VDSK. (BAT transfers data to/from the BATCH file while VDSK allows files to be used as 'pseudo devices'.)

The Solution: Given that the handler space and table sizes are fixed forever, the only alternative available is to have a convenient method for changing them whenever a handler not currently installed in the system is required. On the PDP-12 system which I just mentioned, for instance, the PTP, PTR and LABL handlers are rarely used, and are never used in conjunction with the DIAL handler. Thus I have traditionally kept several different 'system head files' on the disk, using PIP/Y to create an appropriate system as necessary. Another approach is to keep several different copies of the BUILD program, re-building the system whenever you need a different set of handlers. Both of these methods are unattractive because of the large amount of disk space required: each system head uses 50 blocks, and each copy of BUILD takes up 33. A second difficulty with such an approach is that one loses any modifications made to the current set of handlers whenever one creates a totally new system. For instance, I always remove the initial and final formfeeds from the LPT handler, but everytime I build a new system, there they are again! Thus when I first learned that the new version of the SET program would permit changing individual handlers in the system I was delighted.

The format is: SET HANDLER A=B (or A<B) where the names 'A' and 'B' are the -file names- of the two handlers involved. The 'A' handler is the one currently in the system while the 'B' handler is the one you wish to replace it with. Both files must reside on SYS with a .HN (handler) extension. They are created simply by loading and saving the ordinary .BN file which is used by BUILD. This allows one to quickly

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

create a 'library' of handlers which can be installed on a moments notice for a particular job. SET takes care of updating all the resident and non-resident tables, as well as checking for various conflicts which might develop. The number of entry points, for instance, can be different in the two handlers, so if B has more entry points, it is possible that the 'name' table might overflow. The converse of this is that all entry points in the 'A' handler must have been activated when the system was originally built, since SET checks to see that the names in the file match up with names in the system tables. This is occasionally a nuisance: the TC08 handler, for example, has 8 different entry points, only two of which are usually 'inserted' by BUILD. Thus SET will refuse to replace this handler since the file does not agree with the system. Fortunately there is a simple solution to this problem: location 0 in the .HN file contains (minus) the number of entry points; this value can be easily changed with ODT after loading the binary without having to edit and reassemble the source file.

Another restriction enforced by SET is that the entry point names in 'B' cannot be the same as those in 'A'. This seems to me to be an unnecessary restriction which prevents one from using this convenient method for testing various versions of a new handler. (You can, of course, first replace 'A' with some other handler (B) and then replace 'B' with a second version of 'A' which was saved with a different file name.) There are also some 'bugs' in the current version: (1) SET ignores the file size of the .HN file and instead always reads 3 blocks; thus when I accidentally put a 2 block file at the end of a very full disk it gave a fatal read error. Secondly, SET appears to utilize 'once-only' code so that if you try to .R SET and then attempt to replace several handlers without reloading the program, only the first try will succeed. Hopefully this sort of 'kludgy' coding will be fixed in a future release, but DEC probably intended that SET be invoked only at the CCL level which permits only one operation at a time. (Incidentally, to obtain the version number you must .R SET and then type VERSION in response to the # prompt. Only V3A understands about the SET HANDLER directive.)

A much more serious problem with the current version is that the implementors have used the BSW instruction in two places - neither of which was really essential. Thus in place of the code 'CLL RAR;BSW' they could have just as well used three RTL's to accomplish the same result. This requires only a single additional instruction (for which there was room) and would have eliminated the need for the patch given below. As it happens the code containing this sequence is used to determine which blocks to read and write, which meant that I destroyed a number of files on my PDP12 disk until I realized that this must be the source of the problem. Hopefully DEC will take this hint and fix the next release so that it will work correctly on all machines.

Another annoying oversight is that this version was only written to work with 2-page (12K) system handlers! Thus instead of checking to see whether location 07612 contains a '3' (indicative of a 12K handler), SET blithely assumes this to be the case and proceeds to modify block 66 - assuming that this block contains the copy of the resident device-type table. Thus it is necessary to add quite a lengthy patch to this routine so that it will correctly operate on block 0 in the case of the more common single-page (8K) system handler. I find such practices a little disturbing, for although this version is ostensibly distributed for use only with OS/78-V3 systems, its utility (as I hope I have illustrated) is far wider than that particular market segment, and the additional coding required to make it work properly on all systems is truly insignificant. At any rate, since these patches were developed without benefit of any source files, I suggest that you keep a backup copy of the original version somewhere so that if DEC ever decides to release their own 'fixes' you will have a

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

'clean' copy to work from.

Patches to SET.SV, version 3A (dated 1-JUN-79 or 3-JUL-79), to enable use on non-omnibus machines and/or those having 8K system handlers:

```

        .GET SYS SET
        .ODT
Fixes the 60/ XXXX 0;7106;7006;7006;6212;5460
'RAR;BSW' 12240/ 7110 6202;4060
problem.  13055/ 7110 6202;4060

        12165/ 2400 3111
Fixes the 12401/ 6211 3221
'Block 66' 13111/ XXXX 0;6201;7346;1727;6211;7650;1326;3725
problem.  13121/ XXXX 1725;4724;5711;2400;2406;66;7612
        C
        .SAVE SYS SET
        .SAVE SYS SET (Restores prog. to its orig. loc.)

```

RTS8 NOTES

Peter S. White writes: "I have recently obtained version 3 of RTS/8 and am using it on a 64k 8/A equipped with KT8-A and FPP8/A. Initially I thought that FORTRAN IV was not supported in the OS/8 background because it is interrupt driven. However, although I cannot find mention of this in the manuals I have, FRTS.SV tests to see if it is running under RTS/8 and, if so, reconfigures its handlers. This test is a handy one to know, the TTY flag always seems to be set so the test is:

```

TCF      /Clear the flag
TSF      /Skip if RTS/8

```

(Note: In this case the interrupt features of FORTRAN are disabled. If you have RALF code that links to the FORTRAN interrupt system in your program, it will never be executed because the background support does not emulate the interrupt system. In normal, non-real-time FORTRAN programs this is not problem. - RH)

There is, however, a "bug" in FRTS that is peculiar to the KT8-A machine. This is a CIF CDF 0 at 12466 followed by a number of IOT's which cause the FATAL FF to be set and FRTS is aborted by the OS/8 support task. This can be fixed by removing the offending CIF CDF at 12466, moving the contents of 12467-12476 up one word in memory and reinserting the CIF CDF 0 at 12476. This version of FRTS will work under OS/8 or RTS/8. A similar error occurs at 00717 in LOAD.SV, but I cannot think how to get around this one easily. The offending code is a test for CTRL C, so if you are willing to forego this, changing the KSF at 00717 to a NOP cures the problem.

"I normally run RKA0: as SYS: and RLBO: as DSK:. As supplied, OS8.MA equates SYS: and DSK: and this seems to work - e.g. .DIRECT prints the directory of RKA0:. But the command .COMPILE FTEST.FT does not seem to generate a .RL file anywhere unless an .ASSIGN SYS DSK command has been issued. Probably a new system with DSK = SYS would cure this problem.

"I hope these comments will be of use to some of your readers. I am presently looking into ways of supporting the FPP8/A with OS8.MA and would appreciate hearing from anyone who has done this or is contemplating it."

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

Peter is at: Department of Chemistry, University of New Brunswick, Post Office Box 4400, Fredericton, N.B., Canada E3B 5A3 - phone: (506) 453-4775.

PROGRAMS BY DICK MURPHY

At the Fall Symposium in San Diego Dick Murphy (with DEC in Santa Clara, CA) had several very interesting programs that he put on the software exchange system. The following is taken from his documentation.

ADVENTURE: This version of adventure is based on the RT-11 version by Bob Supnik. It is essentially a recoding of the FORTRAN-IV sources into RALF code. The RALF code has been optimized to fit into 32K. Also, many modifications were made to the code to increase execution speed.

Because it is based on the RT-11 version of ADVENTURE, the following features of the FORTRAN-10 version are not supported:

- 1) MAGIC mode was removed
- 2) The HOURS command was deleted

This program is an update of the original ADVENT submitted. It allows the text to be output in upper or lower case, and has a 128 character keyboard typeahead buffer. It also allows the game to be suspended and resumed later.

RXBSY and RXBNS: COS compatible RX01 floppy handlers for OS/8. These are a system handler and a non-system handler that are fully compatible with OS/8. The system handler is two pages long and follows the conventions for two page system handlers that allows BASIC and FORTRAN to work properly. They allow an OS/8 diskette to hold 658 blocks as opposed to only 494 with the standard OS/8 handler.

The system handler cannot be booted with the conventional RX01 bootstrap. To correct this, a program called RXFIX is run after BUILD to modify the system head bootstrap (block 0) to allow use with the standard bootstrap. It is very important to run this program again to restore block 0 before BUILD is run from the system.

Since these handlers are COS compatible, they can be used to copy files to and from COS-310 diskettes. Also, COS310 FILEX can convert a COS file to an OS/8 file on these diskettes that is readable by OS/8.

WPFLOP: WPFLOP is used to transfer documents from word processing floppy disks to OS/8 media or from OS/8 media to word processing diskettes. The WPS floppy is accessed using the COS compatible floppy handlers (see above).

RTFLOP: RT-11 Floppy Interchange Program. This program is used to manipulate RT-11 floppy disk file structures under OS/8. (RX01 ONLY, RX02 NOT supported). The operations available allow most of the functions of OS/8 PIP to be emulated on the RT-11 disk. These functions include:

1. Creating a file
2. Copying a file from an RT-11 floppy to an OS/8 device (disk, DEctape, TTY: LPT:, etc.)
3. Deleting a file from the RT-11 floppy.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 38 - January 1980

4. Listing the directory of the RT-11 floppy.
5. ZEROing (INITing) the directory of a floppy to an empty state.

The program does not recognize wildcards. Therefore a command such as:

```
*.PA? <*.PA/O
```

is illegal. However, there are two TECO macros distributed with this program which allow one to approximate this function.

These programs seem to be very nicely done utilities. They are all written in assembly language, so they run quite efficiently. I hope to get a chance to experiment with WPFL0P soon. It could be a way to get around some of the configuration limitations of WPS-8 and maybe it would also simplify taking advantage of the WPS-8 communications options for moving OS/8 files over the phone.

EDITX, NUEDIT, ETC.

Several variants of OS/8 EDIT have been come to light lately.

First, a file named NUEDIT was on the software exchange disk at the San Diego Symposium. The following documentation was with it:

.EDIT V12B - OS8

Modifiers: D. Spector

E. Steinberger

M. Shugerman

This version of the symbolic editor is a highly modified version of the original OS8 release. The operation is basically the same as documented in the OS8 Handbook with the following items as major new features:

- Commands can be entered in both upper and lower case
- Line numbers are listed in the L and W commands
- W Write command
 Like List, except the editor will remember the line numbers from the command form "n,mW" in subsequent write commands without arguments.
- Z Zap command
 Like the Jerk command except that no punching of the buffer is done into the output file. The editor searches, deletes, and reads buffers until the proper string is found.
- H Rewind command
 The command rewinds the input to the first character of the first file. The present buffer is not modified and there is no effect on the output file. A rewind cannot be done after the end of file.
- Command decoder /A option
 Adds rubouts after tabs in the output file.

Second, Richard A. Karhuse included the following note in his letter to Professor Scobey (see item on VT8E Software in this issue):

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

"I believe Northwestern University Computer Science Research Lab is responsible for EDITX which was distributed at New Orleans. The OS/8 version has line numbers, accepts lower case commands, and a view command- W, Z and H. we are currently modifying it to buffer and perform CRT deletes or commands.

"I will send anyone interested a copy of the current version or the improved version when it is available. I support DECTape and RX01 floppies."

And finally, Jim van Zee writes about what seems to be a different EDITX:

"Here is a complete list of the changes to EDIT/EDITX to fix the problem reported by Alistair Windram in Issue #30, page 56, and to allow operation in conjunction with the OS/78 spooler symbiont as well. The symbiont patch removes the ability to have ODT breakpoints when running the program. This is the way DEC did it, but I'm sure alternative methods are possible (i.e. loc 176 appears to be free???)". The version number (locs 2372-3) is now V13B.

"EDITX is a slightly modified version of EDIT which provides a number of essential improvements, such as lower case commands, output control (CTRL/S,Q), and the elimination of all those stupid linefeeds! While EDIT is certainly not a very good editor, lots of people DO use it, and with these improvements it is actually possible to manipulate textual material such as this letter without going insane! The ambitious soul can easily use ODT (or FUTIL) to convert OS/8/78 EDIT (or earlier versions) into EDITX. The next page contains a OCOMP listing of the differences between versions. Credit for these patches is due to Aaron S. Weg of Data Systems, Inc., 6 Pinewood Drive, Monsey, New York 10952 - phone: (212) 737-3748.

"The comparison below was made with OCOMP and then edited on a VT/78 system (using EDITX) - with the spooler running all the while in background (foreground?). So it works! Incidentally, this system was operating with my new byte-mode system handler, which provides 40% more file space (611 free blocks vs. 438) on the very same diskettes! I had to patch the spooler program so that it could access files on either byte-mode or regular disks, but aside from this (and a patch to FORTRAN-IV), everything else is exactly the same - the same bootstrap, the same duplication procedure, etc. (Contact JvZ for more information about this handler...)

COMPARISON OF CHANGES TO 'EDITX.SV' (22-NOV-78) TO FIX A PROBLEM WITH THE 'Q' AND 'N' COMMANDS, AS WELL AS TO PERMIT OPERATION IN CONJUNCTION WITH THE SPOOLER SYMBIONT UNDER OS/78. THIS VERSION WAS DISTRIBUTED AT THE NEW ORLEANS DECUS MEETING.

	22-NOV-78 VERSION				15-SEP-79 CHANGES			
ADDR	0	1	2	3	0	1	2	3
00000		6100	7700	7701		6232	5001	6100
00004	7000	7000			7700	7701		
00414		1001				1003		
00470			1003				1005	
00554	1003				1005			
ADDR	0	1	2	3	0	1	2	3
01360	5304				5301			

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

01530			1003				1005	
01544	0002				0004			
01550	0002				0004			
01744			4402				4404	
ADDR	0	1	2	3	0	1	2	3
02014	1357				2774			
02020	3402				3404			
02310			1001					1003
02370			0262					0263
02774	xxxx	xxxx	xxxx		3112	5776	1301	

COMPARISON OF OS/8 V3D 'EDIT' AFTER CHANGES FOR 'EDITX'
IMPROVEMENTS: LOWER CASE COMMANDS, CTRL/S,Q SUPPORT, NO
EXTRA LINEFEEDS, 'Q' BUG FIX AND SYMBIONT SUPPORT. AP-
PLIES TO OS/78 EDIT.SV (V1, V2 V3?) AS WELL, BUT SOME
LOCS. MAY HAVE SLIGHTLY DIFFERENT VALUES TO START WITH.

	EDIT.SV 05-JUL-77				EDITX.SV 15-SEP-79			
ADDR	0	1	2	3	0	1	2	3
00000		6100	7700	7701		6232	5001	6100
00004	7000	7000			7700	7701		
00220	1031				7300			
00274	1111	4327			5675	3104		
ADDR	0	1	2	3	0	1	2	3
00414		1001				1003		
00470			1003				1005	
00554	1003				1005			
ADDR	0	1	2	3	0	1	2	3
01250	3071				3074			
01260			1271					4664
01264	7640	5657	2257	6032	3116	7640	5657	2257
01270	5657	7764			6032	5657		
01360	5304				5301			
ADDR	0	1	2	3	0	1	2	3
01530				1003				1005
01544	0002				0004			
01550	0002				0004			
01744			4402					4404
ADDR	0	1	2	3	0	1	2	3
02014	1301				2774			
02020	3402				3404			
02310			1001					1003
02370			0301	0262			0302	0263
ADDR	0	1	2	3	0	1	2	3
02774	4445	2515	2510		3112	5776	1301	

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

ADDR	0	1	2	3	0	1	2	3
03054				1270				1273
03064			1370	5260			7307	1160
03070	0334	1115	1366	7440	3160	1370	5260	0334
03074	1365	7640	1364	1363	1115	1366	7440	1365
03100	5762	3014	5550	4545	7640	1364	1363	5762
03104	3062	1022	7710	5435	1111	4473	0300	0064
03110	1040	1371	7450	5321	1010	1111	4473	0300
03114	1370	7650	5767	7330	0064	4142	0000	2316
03120	5226	4574	2062	5340	1361	7450	5716	1360
03124	1027	3035	4543	1020	7450	5330	1357	5716
03130	3014	1021	3015	1022	6032	6031	5331	6036
03134	3016	1035	3017	5211	0356	1355	7640	5331
03140	1027	7041			7001	5716		
03154		3665	3647	0034		7757	0177	0004
03160	3262	3242			7774	7764		

EDITX help file:

This is a modified version of OS/8 EDIT which uses ^S and ^Q to control terminal output, and accepts both upper & lower case commands. Deleted characters (except for tabs!) are accounted for when tabbing, and extra linefeeds are suppressed. A bug in the 'Q' command was fixed and changes made to allow use with OS/78 symbiont programs (ODT break-point feature removed).

Submitted (but not created) by: Jim van Zee

OS/8 DEVICE HANDLERS

The following is from Jim van Zee:

Here are some useful handlers which can now be tried out by installing them with the new SET command! I just added a few useful improvements to Stan Vivian's line-printer simulator (now called LPTT, it was called OS8LTY).

LPTT: The LPTT handler is a revision of Stan Vivian's line-printer emulator (DECUS 8-694) which allows it to be modified by using the new '.SET' command. This is very convenient for adjusting the handler for various paper widths (.SET LPT WIDTH N) and lengths (.SET LPT LOC 177). In the second case you will be prompted by -(the current page size) (-77 by default) which you can adjust accordingly. Other 'SET' options will either return an error or be ignored. The default DVC is 66; This can be changed by modifying locations '116' and '131'; Thus '.SET LPT LOC 116=6046' and '.SET LPT LOC 131=6041' will change the device code to the terminal. This cannot be done by using the '.SET LPT DVC 04' command since DVC's less than 30 are illegal (too bad!). If you have the latest version of 'SET' you can use the .SET HANDLER directive to install this handler. Otherwise you will need to build a new system. The LPTT handler is extremely convenient for listing files on a device which does not recognize formfeeds (LA36's, etc.).

LPTX: This routine is a modification and combination of the KL8 terminal handler and the LPT handler to do paging every 56 lines. It also prints in octal on the top of the page page numbers in PAL-8 format pp-n or page pp part n.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 38 - January 1980

LABL: AN OS/8 HANDLER FOR PUNCHING LEGIBLE PAPER TAPES

This handler punches alphanumerics on a paper tape using either the high-speed or the low-speed (Teletype) punch. This provides a very convenient way to add a short label for identification purposes before punching either a binary or an ASCII tape. It may be used by any program which allows 2-page handlers, i.e. PIP, FORTRAN, BASIC, FOCAL, etc. The symbols are constructed on a 4x6 grid with spaces substituted for all non-printing control characters (tabs, carriage returns, etc.)

When called, LABL prints a "^" on the terminal and then waits for a key to be struck before continuing. This is similar to the action of the PTR handler and allows time to turn on the punch. This is especially important in the case of the low-speed punch which must remain off while commands are being entered. When output is completed on the low-speed punch LABL again waits for a keyboard character so that the punch can be turned off; since this is not required by the high-speed punch the second wait is omitted. Typing CTRL/C at any time will return to the OS/8 monitor. Typing CTRL/O will generally return to the calling program. No blank tape is punched either before or after the text so if you wish to separate the legible characters from other material, just include a few extra spaces in in the output file. Generally the PTP handler provides a more than adequate amount of blank tape at the beginning anyway.

The TTY handler can be used to create a label just before punching the tape. Input is terminated by typing a CTRL/Z(^Z) The 'super' (KL8E) TTY: handler recognizes rubouts and is thus somewhat more convenient for this purpose than the older AS33 handler. The following example shows how the binary tape for the LABL handler was punched:

```
.R PIP
*LBL:<TTY:
LABL.BN 15 MAY 1976 OS/8 LEGIBLE LEADER HANDLER^Z
^ (turn on the punch, strike any key on the terminal)
*PTP:<LABL.BN/B
```

LABL is added to the system by running the BUILD program (see the OS/8 handbook for complete instructions). After loading the binary, the 'PRINT' command will show: LABL LS HS. The entry points are for the Low Speed and High Speed punches, respectively. Use the 'INSERT' command to add the appropriate one and then rename it with the 'NAME' command. Refer to the first page of the listing for an example of how to do this.

Acknowledgments: The original work on this handler (and the idea for it) came from Daniel T. Brown. Subsequently other features were included such as the provision for both punches, the turn-on/turn-off feature, and operation on a standard (non 8/e) machine. Previous versions have been called 'TEXT' - the name 'LABL' seemed somewhat more appropriate to many users and has thus been adopted. This revision removes the last 'kludge' and restores some order to the internal arrangement of things.

Restrictions: The keyboard is assumed to generate '8-bit' (DEC standard) code with the 'parity' bit always being set on.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

(Author: Jim van Zee, Dept of Chem, U of W, Seattle, WA 98195)

RX01NS

From Jim van Zee:

"The RX01NS handler is a version I created to see if I could get a handler which would pass the 'FUTIL scan test' on the VT/78. After many many attempts I finally came up with this one which can be scanned from block 0 to block 755 at full speed; it is completely compatible with the original DEC single-density handler, AND has the virtue that it will work on the RX02 whereas DEC's version will not because they used so many kludges in setting bits in the command register that they did themselves in. The new floppy handler, of course, will not only work on either the RX01 or RX02, but will also read double-density floppies which this one will not. On the other hand, it is a two-pager while this is only 1 page.

"This is an improved version of the standard single density, non-system RX01 floppy disk handler. It is exactly equivalent to the DEC version except that it uses a fast divide loop to improve performance on a VT/78. It will also work correctly on a RX02 system, whereas the DEC version will hang! Author: Jim van Zee, Laboratory Data Systems, 10320 Ravenna Ave. N.E. Seattle, WA 98125.

"Contact JvZ for information about a new byte-mode system handler which will boot up with the standard bootstrap, but provides 40% more storage and 75% faster access with the very same diskettes on a RX01!"

RK8ESY

From Jim van Zee:

"This is the improved version of the RK8E system handler written by Dick Murphy of DEC and described in the 12-Bit Newsletter Vol 32, Page 15. It preserves the date during bootup and allows booting the 'backside' (RKBO) in case the system side crashes. Here is the boot for RKBO:

loc	contents
0026	1032
0027	6746
0030	6743
0031	5031
0032	6260

"The bootstraps can also be relocated by 10 if desired, which is easier if you are flipping switches. For example, the boot for SYS:

0040	6743
0041	5041

"I suspect that this mod will only appeal to a very few people, but if you try it once you will see how MUCH more convenient it is to put the boot at location 40 instead of location 30. As I mentioned, I suspect that location 30 was used to be compatible with the RK01 without recognizing that the I/O instructions which made 30 a good choice had now been changed so that 40 was the logical origin."

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980FORTZ

Jim van Zee passed on the following information from Aaron S. Weg about his program FORTZ (see also Mr. Weg's letter and the information on his EDITX elsewhere in this issue).

"The routine 'FORTZ' is probably proprietary to Digital and to anyone who has a valid source license which they could demonstrate to me by sending a copy of the Fortran Compiler from Digital at which point, I would then be willing to send a copy of this routine. Note the following features of 'FORTZ':

- a) A call to a sub-routine is executed much quicker.
- b) Uses much less core than standard Digital call.
- c) GOTO statements use less core.
- d) Mixed Mode is allowed.
- e) INTEGER and REAL statements are acceptable.
- f) Dimensions can be included in a COMMON statement as well as in an INTEGER or REAL statement, as in Fortran IV.

"To equivalence a variable into COMMON, it can be dimensioned with zero items. The compiler will accept it although it is an undocumented feature of it."

Mr. Weg also mentions an RL01 version of his RWDISK routine that accesses an RL01 using all 8 bits of each byte to give 13000 blocks rather than the standard OS/8 format that gives 10000 blocks.

COMPAF

Jim van Zee sent the following information and comment:

"COMPAF is Dave Specter's program, which I have found incredibly useful. It needs some more work, i.e. one would like to have a wild-card input specification so you could limit the comparison to a selected group of files, as well as to compare files with different names (but suspected similar contents), such as *.PA and *.BK, for instance. Also a C.D. option to list all the files which are IDENTICAL is badly needed. Finally, it would be nice to have an indication when similar files had different dates, so that one could fix the date if necessary."

COMPAF.HL: "COMPAF is a program for comparing all the files with similar names on two different devices. It will report any which differ as well as indicating those which contain bad blocks. COMPAF requires 16K of memory and calls the Command Decoder when run. The response to the '*' should be: outdev:<indev1:;indev2:;. All three devices must be specified, and indev1 should be the slower of the two (i.e. DECTape). The author of COMPAF is David Specter, formerly with Digital Equipment Corp."

DROP - RECOVR - COPCOM

Jim van Zee notes that these programs (developed in Prof. W.L. van der Poel's lab in Delft and distributed at the '78 European DECUS meeting in Copenhagen - previously mentioned in the Newsletter) are for advanced users, and while he feels they are not on a par with FUTIL, OCOMP, etc. they are certainly useful. Jim indicates that sources were said to be available and he wonders how we might get a copy that could at least be circulated among interested users.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 Number 38 - January 1980

BASIC V5 SPRS

Jim van Zee submitted the following two SPRs on OS/8 BASIC (version 5) and forwarded copies to me.

- 1) The BLOAD (V5) program distributed with OS/8 V3, V3C, V3D and the device extension kit (as distinguished from the BLOAD which is part of the new 'Commercial' BASIC - V6 and V7) has a problem. The problem is that the CCB which it creates when run with the /K switch can be overwritten by a large string array (I think) so that upon trying to save the program after compiling it you get a 'BAD CCB' error message. This problem has obviously been a long-standing one, and was reported to you in June of 1977 by way of a submission including a demo program which exhibited the difficulty. The SPR was returned with a properly working version of BLOAD and a 'Thank You' for help in locating the problem. Since no notice of this bug or its fix ever appeared in DSN, I had just assumed that subsequent releases would have this problem solved; such appears not to be the case, however, since the version of BLOAD in OS/78-V1 as well as the brand-new version distributed with the device extension kit still produce the same result. I have located the patch with the help of OCOMP and would therefore suggest the following fix:

Version dated July 1977 or earlier	Version dated January 1979
02155/ xxxx 0000	02155/ xxxx 0000
02156/ xxxx 6203	02156/ xxxx 6203
02157/ xxxx 7000	02157/ xxxx 7000
02160/ xxxx 1000	02160/ xxxx 1000
02775/ 2534 2154	02775/ 2520 2154
03027/ 6501 6502	03027/ 6501 6502

You will note that the only difference in these patches is the contents of location 02775 before the patch, hence if one were to simply ignore the original value, the same patch could be used for all versions. The last change raises the version number from '5A' to '5B'.

- 2) BASIC.UF was not reassembled for the OS/8 V3D release. Corrections were reported in the Aug-Sep 78 DSN, but only a garbled listing was shown which did not agree with the ODT patches listed in the same article. In addition this overlay contains two instances of keyboard character checks which do not mask the parity bit and hence fail with the VT/52. Also, it is not certain that all entry points agree with the list given in the OS/8 Handbook, p. 6-126. It is very important that these agree, as this is the reference a new user will go by.

DW8E INFORMATION

We wanted to connect a floppy disk system by DSD (Data Systems Design) via the OMNIBUS Converter with our PDP-12. The OMNIBUS Interface "2131" - available from DSD - works fine with an original OMNIBUS, but not in the DW8E. Regarding the old pre-OMNIBUS 8's the IOT's are created sequentially via IOP1, IOP2 and IOP4. The instruction decoder latches in the DSD 2131 Interface are cleared too early by the signal BTP4.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

Two additional IC's (7400 and 7402) mount spare places on the PC-board, and a number of wire changes solved this problem. Engineering drawings are available from : Wolfgang Leber, c/o Max-Planck-Institut fuer Hirnforschung Neurobiologische Abt., Deutschorden Str. 46, D6000 Frankfurt/Main 71, W - Germany.

The test programm "VEP210" delivered by DSD only works if the "BSW's" instructions are replaced by a corresponding subroutine.

W. Leber

VT8E SOFTWARE

Joseph A. Madden writes:

"At the Spring '75 DECUS meeting in Miami, I gave a paper entitled "Using the VT8E Alphanumeric and Graphic Display with OS/8". In this talk I described a version of SCROLL that runs on the VT8E, patches to the Keyboard Monitor and Command Decoder to allow use of the VT8E-Centronics 306 combination as a console device, a two-page output handler for text (crude), and a graphics mode subroutine for use with PAL8, SABR and FORTRAN II.

"We have used VT8E SCROLL for over five years, and it is still popular with the programmers, secretaries and technicians who use the machine. The original version was written for us by Tom McIntyre. We have had many inquiries and I have referred them to Tom, to whom I sent the revised versions. I could submit the package to the library if Tom has no objections.

"I have heard of VT8E TECO, VT8E chess and FOCAL for VT8E graphics, but I do not know about the status or availability at this time." Joe's address is: James A. Haley Veteran's Hospital, 13000 North 30th Street, Tampa, FL 33612. A reprint of the paper may be available from DECUS in Marlboro.

Richard A. Karhuse wrote to Professor Scobey:

"I noticed in the 12-Bit SIG Newsletter (#37, p10) that you are interested in a VT8/E editor. We have had a VT8/E editor which was derived from DEC's standard OS/8 EDIT. It was also modified to run stand alone, to have line numbers and many other little features added. If you are interested, please contact me at the address below." Richard's address is: Computer Science Research Lab. Northwestern University, Tech B626, 2145 Sheridan Road, Evanston, IL 60201. Phone: (312) 492-5248.

MEMORY EXPANSION

C. E. Steuart Dewar writes: "I've enclosed data sheets on a new pdp8 memory product. This is not a standard add-on memory board. The memory is treated as a I/O device so this board is especially useful for pdp8/e/f/m computers that already have 32k of memory.

"I currently have three dozen systems out in the field using this memory board in a multi-terminal text editing application. The speed gain resulting from the additional memory made a vast difference in the terminal response times. Since many of your newsletter readers have large 8e systems, I thought you might find this information of interest."

The following is an extract from the AUX-16 data sheet. "Boost the memory of your PDP-8/e/a/f/m to 48k or more with auxiliary memory from DISC. DISC's new AUX-16

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
Number 38 - January 1980

memory provides 16k of RAM on a single board. The AUX-16 is addressed through PDP-8 IOTs and is ideally suited for fast access storage and retrieval of program data or overlay segments in large systems.

"All AUX-16 IOT instructions execute in a single memory cycle (1.2 usec on a PDP-8/e). Data storage and retrieval functions are typically MORE efficient with AUX memory than with main memory, especially when accessing the first 4k of auxiliary memory.

"The AUX-16 instruction set is exceptionally convenient for handling sequentially stored data since the XRX and XWX instructions automatically increment the AMA (Auxiliary Memory Address) register after the data transfer.

"The AUX-16 is addressed through IOTs, so there is no possibility of interference with the KM8E or main memory. Also, existing 4k systems can use the AUX-16 without the need for a KM8E module."

For further information and prices, contact Dewar Information Systems Corporation, 221 West Lake Street, Oak Park, Illinois 60302 USA - phone (312) 524-1644.

MICRO PROCESSOR SUPPORT SOFTWARE

I find that I still have a note from D. M. Brockman in the Newsletter folder. It was written in July and I am not sure if it has been covered yet. Dave's company is offering some microprocessor support software that runs on the 8 under OS/8. He has "CALOS-68" - a cross assembler for the 6800, a revised version of "CALOS-80" - a cross assembler for the 8080/85, and "SIMOS-85" - a simulator-emulator for the 8080. Contact Dave at FBE Research Company, Inc., PO Box 68234, Seattle, Washington 98168 for details and prices.

EDUSYSTEM-10

I noticed the following item in the October 79 EDUSIG Newsletter. "I would be happy to supply a paper tape, to anyone who wants it, of a modified EDUSYSTEM-10 (i.e. BASIC). The program runs on a PDP-5 or PDP-8S as well as PDP-8I and PDP-8E and can be supplied with or without a self-started loader (bin) that starts loading from RIM loader.

"I have changed the sequence for deletion of extended functions. Or we can keep LOG and EXP until the last and have more room for a program in core. We also added the PTR command to save the pain and wear on tapes, as well as GET(X) and PUT(X) to increase the joy of living.

"The program has been used with our PDP-5 for several years and as long as the computer keeps plugging away we will continue to use it.

"Peter E. Sturrock, Professor of Chemistry, Georgia Institute of Technology, Atlanta, Georgia 30332."

WPS-8 Report from San Diego Decus Meeting

The San Diego meeting was an exciting time for WPS-8 users. Part of the excitement was generated by the excellent series of meetings. Further excitement was generated by anticipation of good things to come due to the formation of a WPS Working Group within the 12-Bit SIG.

Presentations on Monday afternoon and evening centered around DEC's plans for future versions and hardware for WPS-8 and user "hints and kinks". The product line's dedication to the PDP-8 and, in particular, to making the WD-200 do what everybody thought it would was subtly reaffirmed. Among the new products discussed were a serial LQP and a two-page sheet feeder. RX02 drives will be supported, but in a "single-density mode". Talk of software capabilities such as master document, background list processing, and background communications continued. Following the product line's wise, new policy, no firm dates were given for anything.

User hints and kinks sessions revealed a high level of interest and capabilities among users of WPS-8. Discussion primarily centered around user hints and the simultaneous running of WPS-8 and COS-310 on the WD-200. Among the techniques discussed were list processing hints for avoiding blank lines, efficient use of libraries, and use of "ghost terminals" for background list processing and communications.

The sessions dealing with WPS-8 were somewhat unique for DECUS in that the participants were not only the traditional computer expert/OEM, but also relatively new end users with little or no formal computer training. Due to the very high level of grass-roots support in this group of users, the 12-Bit SIG made a significant commitment to these members. This commitment includes specific programs at future DECUS meetings for these people who have formerly been completely outside the structure of DECUS.

The first step in this commitment was the formation of a WPS-8 Working Group within the 12-Bit SIG. This formation of the WPS Working Group should be exciting to WPS-8 users for two reasons. First, there is now a formal organization with a definite commitment to the WPS-8 user who may or may not be the traditional, technical DECUS participant. Second, we have reasons to expect very strong corporate support from the Word Processing Product Line for this group. Preliminary meetings held in San Diego with the product line indicated a high level of interest and produced a number of preliminary plans. This interface with the product line is actively in the process of being developed, and will be further discussed and explained in this newsletter when finalized.

For now, this group is actively working toward Spring DECUS in Chicago with hopefully a greatly expanded formal program oriented toward WPS and the non-technical user.

Steven A. Taylor
WPS-8 Working Group

LAWRENCE H. EISENBERG
COS-310 Working Group
17141 Nance Street
Encino, California 91316
(213) 788-0354

ANNOUNCEMENTS:

ERRATA. In the last issue of 12-BIT (#37) we published some hints and kinks with respect to word processing. Among the hints was a note dealing with the procedure to remove an extra line, where there is a field on that line but no information is available for the field and, accordingly, it is desirable to omit the extra line during final output. The solution provided, while it will work where the exact size of the particular field is known, will not work in all cases. Accordingly, we are providing, in this issue, a fairly exhaustive discussion of hints and kinks for word processing users. Among those provided is a very thorough explanation of the line deletion procedure, which will work for all applications where there is nothing to be printed on a line. For many of you who do use WPS-8, WPS-11 or WORD-11 you may find that the hints which we are setting forth will be sufficiently useful to append to your own word processing manuals.

WINTER 1978 SYMPOSIUM. The San Diego DECUS Symposium simply was superb! It presented one of those rare occasions where everything seemed to go just right. With the guidance of your Steering Committee loyal leaders, Bob Hassinger and Jonathan Lockwood, who also served and guided the Symposia Committee with the selection of sessions, there was hardly a conflict between any two sessions of interest to our members.

Members of both the 12-BIT SIG and the DEBUG SIG steering committees met together on an informal basis, and the common interests of the two groups were explored with the prospect of even more coordination in the future.

For several of us, the Symposium started a day early with a Leadership Training Seminar presented by DECUS (who footed the bill for the extra day). While there were mixed reactions to the full-day seminar (on Sunday, yet), the overwhelming reaction was not only favorable, but fairly enthusiastic. If nothing else, an opportunity was presented to guide new leaders through the maze of DEC's management lines of corporate responsibility and familiarity with DECUS, itself. Among the presentations was a very well received and enjoyed (if not somewhat overly loud) motion picture dealing with the entire concept of "Meetings" -- e.g., why a meeting; how to put one together; how to organize the agenda; control over the meeting; etc. The film, while delightfully humorous, was most informative. Hopefully we can get some more information on how to get a copy and from where.

Perhaps of significant consequence was the fact that the ratio of management and end users to technical personnel attendees was much greater than in any prior Symposium. As a result, many of the end-user "gripes" were fairly heatedly expressed and some DEC people may have been caught off guard. Certainly, several were observed licking some fairly serious wounds.

Certainly the purpose of DECUS is not to provide a forum for "gripes," but on the contrary is intended as an informative and educational opportunity. It was, however,

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

most informative and educational to observe the rising tide of customer awareness of so obvious a lack of awareness on the part of DEC's product line management of end user frustrations and lack of software support. One thing seems certain now. DEC has become aware of the problems facing the commercial end users and has promised to attend to the problem as a priority matter. [It certainly ought to be considered a priority matter and we expect to hear much more both before and at the Spring 1980 DECUS in Chicago.]

DIBOL STANDARDS COMMITTEE MOVES ALONG

Ray Arsonault [DEC] reported on the successful formation of a DIBOL STANDARDS WORKING GROUP. The group, consisting of three committees (Technical, Advisory and Executive) have already commenced their work. Present emphasis is upon DIBOL-11 with a goal to provide a total working compatibility standard from the low-end elevens all the way up. Ray indicated that little, if anything, would be done with DIBOL-8 (from a Standards approach), since it is defined quite well, already. [However, there have been some hints that future editions of DIBOL-8 may take on many of the features of DIBOL-11 and may even be upward compatible.]

It was pointed out that there are over 10,000 Dibol Installations currently in use (from VT-78s through 11/70s). Further studies have indicated that over \$20,000,000 has been invested into DIBOL applications software by customers and OEM working groups. (We can't help but wonder whether that figure includes our own time, which seems to have approached that figure already.) As a consequence, DIBOL has become a MAJOR language within DEC and is being treated as such.

Ray pointed out that at least to this time, DIBOL has not been made available outside of DEC or for use on other equipment. It seems to us, however, that market competitive situations may compel other considerations. One example which comes to mind is the Heathkit computer, which is based on the 11. We understand that although DEC would not consent to license of DIBOL to Heathkit that a compatible language has been purchased for use with the Heathkit, and that CTS-300 software will operate on it. (We also understand that there may be some litigation arising from this situation, but as we have observed in other publications, such litigation is unlikely to proceed far, and is unlikely to succeed if pursued.)

COS-310 VERSION 8.02 AND 9

Mike Daugherty, who has always been one of DEC's most able statesmen at DECUS discussions on DIBOL, and who seems to have a never ending appetite for producing new and better refinements to DIBOL-8, again presented what must be described as an outstanding discussion and presentation regarding COS-310. (Mike participated in many of the DIBOL related sessions, and seemed to be available to most everyone who had questions and problems. Perhaps DEC should re-define Mike's job title to that of Good Will Ambassador.)

Version 8.02 was described by Mike as being available for 64KB DEC STATION 88s utilizing the new RL02 drives. (These disks, which utilize the same drives as the RL01s -- but are not compatible -- provide 20+ MB storage, each, at a relatively small increase in price. Version 8.02 will support the RX01, RX02, RL01, RL02 and RK05 drives. The disk drives cannot, however, be supported together (i.e., you cannot operate a RL02 and RK05 at the same time). Further support is provided for the LA34,

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

LA38, LA35, LA36, and LA18-ES printers. Other than the handlers, there were no changes in Version 8.02, except that the binary scratch area was reduced by 2 blocks to allow for the the new drivers. This version is now available from DEC.

Numerous possible improvements were discussed with respect to Version 9 which is contemplated for sometime within the next year. We saw a demonstration of an experimental version and found some very impressionable features. [Remember, none of the features is promised, although demonstrated.]

First, the operating system has been modified to provide for fast execution and reduced disk space. The ridiculous 8K buffer for the LQP has been reduced to 4K (which may, or may not, be of any consequence, as low end systems, such as 32K WD-78s, still will require programs using not more than 12K memory in order to operate with the LQP. While we were told that this is not necessarily the case, we have found that no program requiring 16K at compilation time will operate on a 78, with 16KW memory and an LQP. Although this modification is represented as reducing the overhead to the same as that which is required for all other printers, we also were informed that an additional 2KW will be required with respect to other operating features of the new Version. Until we receive a copy of Version 9, we will be unable to comment further upon the memory limitations, if any (unless, of course, DEC would like to advise us of its final conclusions on this point).

Other truly significant improvements include: (a) being able to lock out CONTROL C from an operating program; (b) ability to mix any 2 disk handlers (but not an RL02 and RK05); (c) source scratch area to/from disk as required; (d) comments in BATCH files; (e) Binary scratch area in a named file; (f) SET commands; (g) wild card PIP; and several other features, including one described by Mike as "SUBSTITUTE" (which is to remain a secret for the time being). [Our personal spies have informed us, however, that this command will be available as a TEXT EDITOR and/or to permit changing some function names (e.g., change "FINI" to "CLOSE", etc.).]

One of the other features also highlighted a current problem which often is overlooked by all of us. Under all current versions of COS-310, it is necessary for the COMP file to be at the very top of the Directory. If it is not, it is possible for the compiler to destroy considerable program detail as it seeks out buffer space areas to compile programs. Version 9 will eliminate the problem and the requirement that COMP be at the top of the directory.

By no means is this report exhaustive of the features which were described for the new Version (again, all considered possibilities and subject to change). We will discuss some of the other features in a future issue and, with luck, may be able to be more specific by actual trial of the current experimental version. [Look, Mike, we did spell your name correctly this time, so let's see what we can do on this!]

DEC'S WORD PROCESSING SUPPORT -- OR LACK OF SUPPORT

Undoubtedly the most profound announcement -- and the most debatable -- came from Bob McGeary of the Word Processing Product Line. On the up side, he announced a departure from prior statements regarding support of the RX02. WPS-8 will now support the RX02. [Subsequently we were informed that Version 3.02 is now available and that it contains the RX02 handler.] However, WPS, itself, will operate only in the RX01 mode on the RX02 drives.

The most controversial statement announced by Mr. McGeary was that the WPS-8 product line will not provide future support to 310s and 100 series (101s and 102s) and will provide only limited future support to WS-78 series! Following considerable debate on this point, the matter remains unresolved to this time, with some DEC management suggesting that Mr. McGeary was incorrect and others saying he is right.

This type of situation simply is intolerable. DEC has made numerous promises to the users and purchasers of this equipment, regarding future enhancements. These promises came from the very top, and for DEC to now announce contrary plans can only seriously damage DEC's reputation in the entire community. We sincerely hope that management gets its act together and authorizes us to publish contrary information (but to be followed by concrete action). DEC's word processing on the EIGHT has not been improved for well over a year. The poor operating level of the 200 clearly shares the responsibility, but it should not result in setting DEC's Word Processing System so far behind other competitors.

We certainly hope that the Word Processing Line will get its act together and recognize the real problems involved here. There is no question in this writer's mind (from personal comparisons) that DEC's WPS-8 has potential to be the finest possible system available. To not develop this potential, and to permit the competition to surpass DEC's system, is most unfortunate.

In view of the long article we are enclosing with respect to Word Processing, we will reserve additional comments upon the WPS announcements, and the 200 problems and limitations to next issue.

NEXT ISSUE

In our next Newsletter we will discuss the WISH LIST presented at the Symposium and several helpful hints and kinks with respect to DIBOL. As always, your contributions would be appreciated, and certainly will be published. We are especially looking for interesting subroutines which other readers can share.

WORD PROCESSING WITH DEC COMPUTERS HINTS AND KINKS

LIST PROCESSING HINTS

FIELD IDENTIFIERS AND DATA PROCESSING

The "lists" which are developed in list processing often are useful for data processing activities as well as many of the Word Processing and List Processing purposes. For PDP-11 users, many of the data files developed under Word Processing may be addressed directly by data processing. However, for PDP-8 users the WPS-8 files (which are saved in a format similar to COS-310) cannot be addressed directly by COS-310 or OS/8. While the WS-200 series provide for direct communication between Word Processing and COS-310, conversion often still is required. Thus, WPS-8 files must be converted in some manner. (Conversion utilities for both COS-310 and OS/8 are available through DECUS LIBRARY. These utilities transfer list processing type files between the various systems. The conversion procedures are not discussed in this paper.)

It is most helpful, therefore, to maintain the LIST FIELD IDENTIFIERS as upper case characters. While the DEC WPS manuals show the field identifiers (e.g. - <field1>) as lower case fields, such was not meant to be a required form for identifying the fields. The use of lower case by DEC was a throwback to computer manuals which used lower case to indicate operator decisions, as opposed to upper case which indicated mandatory acts.

Since each of the WPS-8 systems utilizes special characters to indicate lower (and upper) case shifts, any conversion program is going to require considerable additional (and wasted) time in order to perform the conversion, as each of the special characters will have to be stripped from the field before the data can be used by the data processing system.

If there is even the remotest possibility that your list files will be used in data processing, it is important to avoid the use of hard [the RETURN key] returns except at the end of a field identifier. In other words use one identifier for every line of text. For example:

DO NOT USE

```
<NAME>John Doe
<ADDRES>123 Any Street
Our Town, U.S.A.
00123
```

DO USE

```
<NAME>John Doe
<ADDS1>123 Any Street
<ADDS2>Our Town, U.S.A.
<ADDS3>
<ADDS4>
<ZIP>00123
```

In many conversion programs, and nearly all data processing programs, the carrier returns within a field will be read as a terminator, and the information following the return will be lost during the conversion or use by the program.

While the use of several fields may appear somewhat cumbersome at first, the benefits soon become very apparent. Also, the more available fields, the easier it is to edit and to SORT!

SELECTION SPECIFICATION - TO SELECT ONLY IF SOME CHARACTER EXISTS

The DEC manuals fail to disclose the selection specification which can be used to select a record only if a field has information. The wild card specifications presented by DEC are ? and *. The ? is used to replace a letter (i.e., it must be preceded or followed by some other character other than a ?). The * is used to define a field as containing ANY OR NO characters.

From time to time it is necessary to select a record ONLY IF A GIVEN FIELD HAS SOME INFORMATION. There are two possibilities; the first example given is the most

reliable:

- (1) if<field5> =?*
then process record
- (2) not if<field5> =
then process record

(The use of lower case is for example, only. USE UPPER CASE!) The last example would follow other qualifiers; if used alone the results are not predictable.

DELETING UNUSED LINES FROM FINAL OUTPUT WHERE THERE IS NO DATA

Regretably this author allowed an article to be published upon this subject (12-BIT Nov. 1979) which contained an inaccuracy. The information which is presented in this paper is correct, and has been tested under several conditions. (The prior article presented a situation which would work only if the field size was known in advance.)

DEFINING THE PROBLEM: EMPTY FIELDS ON LINES WHICH SHOULD NOT BE PRINTED. The problem which often is encountered is how to eliminate blank lines which are printed when there is a field which is empty, but which has been defined in the form. We will use an address block as an example.

<NAME>
<TITLE>
<COMPANY>
<APT/SUITE#>
<ADDR1>
<ADDR2>
<ADDR3>
<CI/ST/ZP>
<DROP>

In the example presented it is obvious that several of the fields might not be present in the final printout. The individual may have no title; s/he may not be associated with a company; there may be no apartment or suite number; there may only be a single address line. However, if the FORM is created in the manner indicated, which, in the example (and only by way of illustration) would be the same as the LIST, the final output would be printed with blank lines for each line on which there is missing data.

There is a solution. It takes a little planning, but once understood, it is simple to apply to every situation. (Just keep in mind, however, that this solution will cause each missing field to disappear and to bring the following line up one line feed! You must remember to allow for this, if the missing lines could affect other line-count features of your form.)

The first step is in the creation of a FORM. To accomplish the desired result for any set of circumstances it is necessary to create two FORMS. The first FORM should include only the variable information, and will, itself, become the LIST which then will be used to create the actual FORM or PRINTOUT. **THERE CAN BE NO SPACES OR TABS ON ANY LINE WHICH MAY "DISAPPEAR", EITHER IN THE ORIGINAL LIST OR ON THE FORM.** (Adjust the Left Ruler in lieu of a single tab, if indentation is desired.)

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

The FIRST FORM is created to determine which, if any, fields are not present and automatically to create a "wrap", as opposed to a HARD RETURN, for each such field. It also is used to create the second LIST. To accomplish this, it is necessary to create "soft" returns on each line which may not have information upon a field. This is done by using dummy rulers after each line which reasonably is expected to "wrap". Using the LIST above, and assuming that EVERY LINE may possibly have a missing field, we could create a form as follows [NOTE THE RULERS!]:

```
L-----R-----
. 1 . 2 . 3 . 4 . 5 . 6 . 7 . 8
.....0.....0.....0.....0.....0.....0.....0.....0.....0
```

```
<<NAME><NAME>
LT-----R-----
<<TITLE><TITLE>
L-----R-----
<<COMPANY><COMPANY>
LT-----R-----
<<APT/SUITE#><APT/SUITE#>
L-----R-----
<<ADDR1><ADDR1>
LT-----R-----
<<ADDR2><ADDR2>
L-----R-----
<<ADDR3><ADDR3>
LT-----R-----
<<CI/ST/ZP><CI/ST/ZP>
-----L-----R-----
<<DROP><DROP>
LT-----R-----
<<>
```

Note that each of the rulers is identical, except for the dummy tab which follows every alternate ruler. The only purpose for the tab is to create a new ruler which can be imbedded. (If the rulers were identical, they would all disappear, and the method described could not be used.) Also note that the last line, DROP, has been indented by changing the left margin. The "indent" feature may be used on any line and is used to avoid the insertion of tabs or spaces which necessarily will defeat this utility. Also note the "<<>" identifier to create the new list! (Down arrows indicate hard returns which may be observed with GOLD VIEW.)

Proceed to the beginning of each line AFTER A LINE WHICH MIGHT BE EXPECTED TO HAVE AN EMPTY FIELD. (This could be every line.) Use the Blue LINE key to travel from line to line. Strike the RUB CHAR OUT key ONCE. (This will delete the hard return and, upon a GOLD VIEW, will disclose a funny looking circle at the end of the sentence, instead of a down arrow, but it will not affect any of the characters.) Repeat for each line which might reasonably be expected to have an empty field. [If it becomes necessary to edit the last letter, back the cursor to the end of the line -- this will place it under the last letter -- and insert the new characters. The last letter will continue to travel and, if undesired, must be deleted.]

Run the List Processing feature, creating a document. The document created by this feature will, itself, become the LIST for the second part of the program.

Upon completion, you will have created a form which, when operated with the List Processing Feature, will result in a new LIST which will have "wraps" in each empty field, between a ruler. YOU MUST BE CAREFUL TO AVOID TABS OR SPACES IN THE FIELDS AND IN THE FORMS or this utility will not work properly.

The SECOND STEP is to create another FORM, which is identical to the first, except for the special field creators.

From the following illustration note that that extra field identifiers have been removed. This will be the final list and will eliminate the spaces between lines which otherwise would have been created as a result of unwanted fields.

If you should find spaces between lines, the problem most likely will be that tabs or spaces were imbedded in either the FORMS or the original LISTS. Check them carefully.

The following form is such an illustration:

```
L-----R-----
. 1 . 2 . 3 . 4 . 5 . 6 . 7 . 8
.....0.....0.....0.....0.....0.....0.....0.....0

<NAME>
LT-----R-----
<TITLE>
L-----R-----
<COMPANY>
LT-----R-----
<APT/SUITE#>
L-----R-----
<ADDR1>
LT-----R-----
<ADDR2>
L-----R-----
<ADDR3>
LT-----R-----
<CI/ST/ZP>
-----L-----R-----
                <DROP>
LT-----R-----
```

As with the first FORM, line feed to the beginning of each line AFTER the field which may not be present, and enter a RUB CHAR OUT to delete the hard return. (If you merely copied the document, be careful, as you may delete a character from the preceding line. To edit this problem, BACK UP to the preceding line (you will be on the last character). Re-type the last character (the one which is above the cursor) and any character which was deleted. Finish with a hard return. Delete the remaining character above the cursor, which should remove the hard return, also. (Check with GOLD VIEW.)

Now, USING THE NEW LIST CREATED BY THE LAST FORM AS YOUR LIST DOCUMENT, run the list processing again. This time, the new document (which can also be a direct PRINT) will cause all of the empty fields to "fold" upon themselves, so that all of the rulers

with "soft" returns will collapse and the final output will be without line spaces between information. While each of the rulers will appear on the screen, there will be no returns within them and the printer will skip to the next line of text.

If it appears that there is a space between rulers on which there was no data, check to see if there had been a space or tab on either of the FORMS or LISTS used for the procedure. Check your original LIST with GOLD VIEW. Each empty field's right arrow should be followed by an immediate down arrow.

Remember, you only have to create the two forms ONCE. They can be used for every processing run. (Actually, you only need create the form once, and then add the extra field identifiers to one of the forms. If you should get a line wrap, because of the extra space required by the new field identifiers, don't worry. The program will automatically adjust.

PROGRAMMING NOTE: Although you can use the same selection specification for both forms, you also can use the simple specification of "process record" for the second run, as you already have specified the records to be used.

COMFORT NOTE: Although this may appear somewhat clumsy, it actually is rather easy and once you get the hang of it, you will find the procedure very useful!

USING LIST PROCESSING TO CREATE AN INDEX OR TABLE OF CONTENTS

Presently there is little ease with which to create an index or a table of contents with the existing WPS-8 or WPS-11 systems. While WORD-11 will semi-automatically create an index and Table of Contents, and most dedicated word processing systems do the same, some ingenuity is required to do the same with DEC's systems (although we are assured that this, too, will change some day!).

For the time being, a fairly long document can become a LIST document using the following procedure.

First, copy the document over to another location (or on another diskette), as you are going to alter it considerably.

Second, decide on some easy shorthand for the categories you are going to use with your index or table of contents. For example, you might wish to use <H> for headers; <N> for names, etc. Choose a character to be used as a dummy field identifier, e.g. <X>.

Enter a terminator and the dummy field identifier in the PASTE buffer, as you will be using it quite a bit during this exercise. (To enter it in the paste buffer, type it and then cut it.)

E.g.: <><X>

Start the document with the dummy field (e.g., <X>) and proceed to the first data which is to be used in the Table of Contents or Index. Let's suppose the first data is a header, which will use the <H> identifier. Enter a terminator <> and field identifier <H> immediately preceding the header and then enter the PASTE immediately

after the header. Thus, the document would appear something like this:

```
|----- t o p o f p a g e -----|
|<X>|
|  |
| (miscellaneous data) |
|  |
| <><H>TITLE OF DOCUMENT<><X> |
|  |
|<><H>First Subheading<><X> |
|  |
| (miscellaneous data) |
|   <><X><N>(desired name)<><X> |
| (miscellaneous data) |
|  |
|<> [entered as last character in document] |
|----- b o t t o m o f p a g e -----|
```

In the same manner, identify the different titles throughout the document, such as names, subtitles, books, etc., until you have identified each item which will be used in your index or Table of Contents.

CAUTION: As you proceed through the document, enter the PASTE in a random manner (i.e., insert the dummy field identifier <><X>) about every 2/3 screen, or more often. This is necessary as no field may contain more than 1500 characters, and to avoid an error message you will have to insert the dummy field every so often. It doesn't matter how often you use the dummy field, as it never will be referenced during list processing.

At the very end of the document, be sure to enter a terminator <> or an error message will occur (it won't affect your program, but no error is more comforting than some buzz error which might leave some doubt).

After proceeding through the entire document, you can create a very simple FORM and SELECTION SPECIFICATION. The FORM may consist of a single entry (e.g., <H>). The selection specification may be "process record". Operating the List Processing, then, will transfer each data identified with the <H>, and will skip all of the rest. (If you have to format the output, it will be much easier to do so after running the list processing.

Also, if you have the type of document which might require some form of sorting, such as alphabetical listings, you can perform some minimal alphabetical sorting by use of the wild cards in your selection specification. (This will require several runs through the list processing; e.g.: if <N>=A* then process record, will pick up every name starting with an upper case A, etc.) If there are only a few records, then use of the cut and paste feature will probably result in an easier, as well as faster, processing.

Another feature, which will result in much faster operation if several field identifiers are being used, is to utilize the double LIST feature (i.e., create a new LIST with a single pass). To create a new LIST, set up your FORM (for the above example)

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

as follows:

```
<<H><H>
<<N><N>
<<>
```

Processing the entire document will fill a new document with each field, in a random manner, and you then can run a second pass which will be more selective as to the order in which you want the items to appear. All of the dummy <X> field data will be omitted from the new LIST.

USING LIST PROCESSING TO LOCATE DISKETTE INDEX INFORMATION

It is not unusual to want to find information from a diskette index, and to avoid going into the index (where there is a danger of losing the document). The diskette index is set up as a LIST document, and can be used for many purposes. (It even can be alphabetized, or otherwise sorted, if care is used, by using the procedures set forth above, or with the SORT program available to WPS-8 users.)

The INDEX LIST for each diskette is formatted:

```
<n>title data <#>5<>
```

Using <n> for your selection specification, you can seek any type of sequence desired. E.g., if you want to find if "John Doe" appears, and on which document(s), the selection specification could be set:

```
if <n> =<*>John Doe<*>
then process record
```

The FORM would be set:

```
<n> <#>
```

Running the List Processing will show the presence of the requested data, and the document number, each time it appears. On long indexes this can save time, and is more positive than using the I command to examine every page of the index.

LINE NUMBERING USING WORD PROCESSING

Presently there is no easy way to number the lines on a document under WPS-8. Perhaps some day the powers to be will provide us with this feature, but for the time being it is necessary to use some planning in order to accomplish line numbering.

At the moment, the easiest way to number lines, whether starting with 1 and proceeding to xxxxx, or repeating the same number of lines per page, is to do it by brute force.

Create your document in the normal manner, but allow sufficient extra space on the left margin ruler for the numbers to be used plus at least two spaces. Thus, if you ordinarily would use the left margin for your left ruler and expect to use three digits for the numbers, set your left ruler, initially, five spaces to the right.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
 NUMBER 38 JANUARY 1980

(NOTE: There will be a slight variance in this procedure for inside paragraphs. This is discussed below.)

Upon completion of the document, AND AFTER FINAL EDITING, the line numbers can be added by re-setting the left margin on the ruler to its normal location AND INSERTING A TAB AT THE FORMER LEFT MARGIN LOCATION. While this would ordinarily cause the text to "re-wrap", it will make no difference. Proceed to the beginning of each line, using the BLUE LINE editor key.

Enter the line number and then TAB. Repeat this for each line to be numbered. Since the text already has been edited, the new line numbers will not affect your prior formatting, as you are using all of the extra space with the line numbers and tabs.

If you need to enter identical line numbers for each page (e.g., such as with legal pleadings of 1 through 28 or 32 for each page) then you can do this with a User Defined Key. Anything more, however, will use up all of the buffer space available for the User Defined Keys.

The use of the line numbers and tabs will not affect right justification, as each line number will follow a soft return. HOWEVER, SUBSEQUENT EDITING WILL BE VERY DIFFICULT. Therefore, try to avoid numbering the lines until the document is ready for final output.

INSIDE PARAGRAPHS

To use the line numbering feature on inside paragraphs, where the numbering is to remain on the left margin, use a W (wrap) in the ruler instead of the L for Left Margin. The first line of each inside paragraph will have to be double tabbed, but you will find it fairly easy to master after a few attempts. When you are ready to insert the line numbers, it will be necessary to remove the W from the ruler, and to replace it with a T (tab). When tabbing over from the number insertions, the text will remain formatted in the same location as with the W, and, as before, right justification will remain undisturbed.

If further editing may be expected, it may be easier to retain a copy of the document before line number inputting, especially where the editing may be extensive. The procedure indicated is not intended as a solution, but, rather, as a procedure which may make life somewhat easier for you.

INSERTION OF PORTIONS OF LONG DOCUMENTS, TOO LONG FOR "CUT AND PASTE", AND/OR WHERE ALL IMBEDDED MATERIALS IS DESIRED, AND/OR WHERE A "GO GET" ROUTINE IS NOT AVAILABLE BECAUSE THERE IS INSUFFICIENT ROOM REMAINING ON THE DISKETTE

It is not at all unusual to have the need to use a portion of a long document in a document presently being created. Quite often, also, the size of the required material exceeds the buffer space allowed with the "cut and paste" method (which often deletes a lot of the material you wanted); the remaining space on the diskette is insufficient to allow you to GO GET the old document, and then cut out the unwanted portions (even if all you want is in the first few pages) or you want to retain imbedded materials, such as rulers and page markers, and the cut and paste method won't retain them. Do not lose hope, there is a fairly simple remedy.

SOLUTION: Edit the old document to the portions desired. Enter a "boilerplate library" type of indicator at the beginning of the text to be copied, and a double arrow terminator. E.g.:

<<COPY1>>text material (may be as long as needed) <>

Use the same procedure for each section to be copied, but identify each portion with different names, e.g.: <<COPY1>>, <<COPY2>>, etc. (These identifiers can be removed, later, quite easily by using the blue <> key to advance through the document and rubbing out the identifiers.)

Note the drive and document number of the old document. Return to your new document and, with the Gold Menu (i.e., the editor menu) feature, change the boilerplate library to the drive and document number containing the old document.

Proceed to the portion of the new document which is to receive the old document's information, enter GOLD LIBRARY and the name (e.g., COPY1, COPY2, etc.). The information will be transferred, including all imbedded materials, such as rulers.

After using this method, be certain to reset the boilerplate library, in the editor menu, for its proper location.

ABBREVIATION AND BOILERPLATE LIBRARIES

There is no end to the utility to which the system libraries may be appreciated by the Word Processing operator. Undoubtedly, these features are the most important individual assets of the entire system.

Naturally, the needs of each user will be different. We believe that the following hints will be of interest to most users.

UPPER VS. LOWER CASE FOR FIELD IDENTIFIERS.

Again, as with List Processing, there is no requirement that you use lower case field identifiers for the libraries. In fact, upper case identifiers generally are much preferable, as reference to the library document may be made in upper or lower case and still retrieve the document, whereas if the library field identifier is in lower case, only a lower case identifier will retrieve it. This especially can be annoying if you are seeking an abbreviation library document (which does not echo the input on the screen) and you happen to have the caps lock activated.

LOCATION OF LIBRARIES

The Word Processing manuals and the self-paced teaching manuals for WPS-8 identify SYSTEM 2 and SYSTEM 5 as the location for the abbreviation and boilerplate libraries. Indeed, all the software for the Word Processing software comes with SYSTEM 2 and SYSTEM 5 initiated as the respective libraries.

There is no magic in the assignment of locations for the libraries and your own particular needs should dictate where these libraries are located, and even whether you might wish to change libraries during different operations (a very helpful and powerful feature).

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

In a client or job oriented operation, where each client or job is assigned an individual data diskette (or RL01 allocation) it might be most helpful to always have the boilerplate library as the first document to be created on that data diskette (which will always be document #2, as #1 is reserved for the diskette's index). If this is done, data which is repetitious for each client or job easily may be recalled by using the same abbreviations or identifiers for each diskette. For example, in our own operations we would identify the name and address block of our client with a field identifier of <<CLIENT>>. Since this information resides only on the diskette in use, every time the library identifier of CLIENT is used, the name and address of that client is displayed in the document. Naturally, the same is true with all information which applies to the specific account, but which is similarly identified for all accounts.

In this manner, the SYSTEM diskette's space is reserved for other needs, and many other libraries.

ALTERNATING LIBRARIES

There is no particular requirement that the library document always be in the same location. On the other hand, it often is helpful to be able to have several documents available on a given diskette which can be utilized as a library document for a particular purpose. This is especially helpful in creating new documents, where there is going to be repetitious use of some phrases. A new abbreviation library can be created, for these phrases only, and the phrases called with short entries. When completed, the library contents can be deleted (or retained, if desired) and the library document changed to the standard.

The use of such a "temporary" library is especially appreciated when one no longer has to search through the current document for specific phrases to be "cut and pasted" at a specific location.

Also, if a library document becomes too lengthy, then it takes a considerable period of time for the computer to find the phrases you need. To avoid this problem, you often can break your library documents into categories, and, knowing the category desired, assign that document as the library (abbreviation or boiler plate) document for the current assignment.

USE OF THE HELP COMMAND FOR LIBRARY CONTENTS

As use of library documents increases it becomes increasingly difficult to remember field identifier assignments, and hard copy reminders become antiquated, misplaced, or unhandy. There is, however, an on-line solution, and that is a HELP COMMAND.

When creating a library document, the first field identifier should be <<HE>> for the abbreviation library and <<HELP>> for the boilerplate library. (Entering "help" will call the field in both cases, although the extra letters ("lp") will appear on the screen after an abbreviation library call.)

Prepare a Table of Contents which identifies each field identifier and its meaning, which can be called by the HELP command. As each new abbreviation is added to the library, the HELP section also is updated with the new command information.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

To seek and examine the HELP information, which only can be accomplished while editing a document, the operator simply (1) enters the SELEct key; (2) enters GOLD ABBREVIATION or GOLD LIBRARY and the word HELP (although only HE is required for an abbreviation); and the HELP information is displayed upon the screen. [Reference to "sub-help" libraries may be followed with another GOLD LIBRARY command.] After examining the displayed information, the operator (3) strikes the CUT key and all the displayed information is removed from the screen to the position where the SELEct was inserted and the library may be accessed for the desired field.

By no means is the information provided here exhaustive of the potential for the HELP library. One may use HELP as a key to provide the operator with special instructions with respect to procedures to be followed with specific routines or documents. In fact, one can have a separate library entitled HELP. (To access the library, the library document is changed to [diskette].HELP, from the Editor Menu, which automatically will change the library document.)

SETTING UP THE LIBRARY DOCUMENTS TO DELETE THE HARD RETURN

There are two methods available to avoid hard returns following a Library Document call. (I.e., where special formatting is required the formatting must follow the library field identifier in order to be imbedded.)

The first method, of course, is to have the library information begin immediately after the field identifier (<<field>>Data xxx).

The second utilizes the soft return described in preceding sections. The following is an example:

```
L-----R---
<<DOCUMENT>>
LT-----R---
                TITLE
                (data)
<<
```

Without modification, if DOCUMENT is referenced by the library an extra return will result, as creation of the document necessarily required a hard return after the field identifier.

However, with the insertion of a new ruler (a dummy ruler is indicated, but it should represent a required format) the hard return can be changed to a soft return (by moving to the beginning of the line immediately after the field identifier and striking the RUB CHAR OUT key) and, when referenced, the soft return will be ignored. The imbedded ruler also will appear.

USE OF LIBRARY DOCUMENTS FOR EXTRA RULER STORAGE

Quite often the ten ruler storage availability of the Word Processing System is inadequate, either because more rulers are required or because it is difficult to remember which is which. There is an alternative.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER
NUMBER 38 JANUARY 1980

Using the same technique for removing the hard return described in the foregoing sections, rulers can be saved in a library and can be called by document type. This especially can be helpful for unusual documents, but also is helpful for general documents. The following are examples of two rulers. Expansion of the concept is quite unlimited and, obviously, up to the individual user.

<<LETTER>>

L---P----T-----H-----J-----

<<

<<SCHEDULE>>

L---T---T---<----->R

<<

Naturally, a schedule of all of the rulers can be part of the HELP library.

SPREADING A TITLE OR HEADING

One final note. The soft return also can be used to spread a word, or series of words, across an entire page. Provided that the RIGHT MARGIN is set with a J (justification), everything on the soft return line will be spread across the page if the next line commences with any type of an imbedded command (e.g., ruler, print control, page marker, etc.). Example:

L---P----T----T-----H-----J-----

(Miscellaneous text to the next line)

T I T L E

L---T---T---T-----H-----J-----

By creating a soft return (i.e., using the RUB CHAR OUT at the beginning of the first line after "TITLE" in the above example, the word "TITLE" will be spread across the page as follows:

T I T L E

In order to create the "soft" return, it only was necessary to modify some portion of the ruler, imbed it and then delete the hard return with the RUB CHAR OUT from the beginning of the line.

SUMMARY

DEC's Word Processing Systems (and even those which utilize DEC equipment) clearly are among the most powerful available on the market today. The potential -- indeed the need -- for improvements is all too obvious, if DEC intends to remain a serious contender for the Word Processing Market.

In the meantime, there are numerous routines which are available in the existing system which can make it work better and faster for you, and that is what automated word processing is supposed to be all about.

The examples provided here are but a few of the many work saving features which are available. It appears that these examples have never previously been documented by DEC, which really is a shame.

We do hope, however, that the foregoing will be of some assistance to the Word Processor user and that this Article may become part of your Word Processing Manuals.

SEE YOU IN CHICAGO!

Don't forget, the next Symposium is April 22-25 in Chicago. We doo look forward to seeing you there.

- - - 3 0 - - -

COLLEGE OF TECHNOLOGY College Square East, Belfast BT1 6DJ

Principal: W.F.K. Kerr, MSc, PhD, CEng, FIMechE,

Telephone 27244



Department of Mathematics and Computer Studies

Head of Department: G.E. Robinson, BA, AFIMA

Our college offers a wide range of courses involving computing topics. One of these courses involves programming in COBOL and RPG. At present programs in these languages are run on computer systems outside the college. This situation is not completely satisfactory and we would like, if possible, our own facilities.

To satisfy this need we would like to acquire COBOL and/or RPG compilers for our PDP 8 system. Do you have information on any PDP 8 systems that run such compilers? If so could we have details, especially price. We would hope that they would not be too expensive.

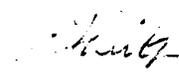
I have details of ETOS but we have decided not to buy it. At present we run OS/8 BASIC, FORTRAN IV and CESIL in BATCH mode under OS/8, interactive BASIC under EDU25 with 8 users and DIBOL under COS 300. Most of the schools in our area will now either have their own micro-computer or have access to one. Hence the only advantage of ETOS to us would be COBOL and we feel this is an expensive way to buy COBOL.

All the colleges and schools in the area are getting ZILOG Z80 micro computer systems and we hope to have COBOL available on these. But for BATCH work and card input we would prefer to use the PDP 8.

Our system is:	PDP 8/E CPU	VDU
	32K store	Mark sense card reader
	Dual RK05 disc drives	Lineprinter
	Dual DEC tapes	7 terminals (4 remote)
	FPP (no double precision)	

Hoping you can be of help.

Yours sincerely,


W. T. Smith
Lecturer in Computing

MUSSETTER REALTY

INCORPORATED

POST OFFICE BOX 838

WILLIAMS COLUSA COUNTY CALIFORNIA 95987

(916) 473-2123

OFFICE ADDRESS:

464-812 STREET

WILLIAMS

December 18, 1979

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

Dear Bob,

There is such a dichotomy between end users, such as myself, and computer wizards who comprised perhaps 95% of the attendees at the fall DECUS meeting in San Diego that I hesitate to contribute. But, you asked for it, so it here goes.

Finances

I would regard it as normal and reasonable to be charged an annual membership fee in the range of \$50.00 to \$100.00. The revenue raised would be claimed by DECUS as such for administrative purposes to the extent of perhaps 10% and the balance would be retained by the SIGs. I would foresee a need for a member to specify one or more SIGs to which he wished to belong; his single fee would cover his membership in both. Membership would entitle each member to:

1. all issues of a single SIG's Newsletter for a year
2. a discount on the Symposia fees
3. software exchange privileges, especially, access to a cross-referencing system (which is to be developed).

If a member specifies more than one SIG, that member would have to pay a bit extra to receive the additional Newsletters.

I think the above format will put wheels under the SIGs and DECUS and if we will go directly to this it will save several confusing revisions. Most organizations operate on the described scheme or something very similar.

Even a non-computer person can see there is much to be done both on technical levels and on educational levels.

Education

Symposia and seminars are, despite the above recommendation regarding membership fees, the most promising source of revenue for DECUS. Gary Cole stated there are 5,000 DEC-Station 78 installations. If I heard him correctly, and if a significant portion of these are operated by people such as me and my people, there is a big teaching

job to be done. Areas of interest to us include:

Applications software

Accounting, Data file management systems such as "Giant" (purchased by DEC during summer 1979 I found out), and/or INFPAK, statistical routines, and financial management programs. A rigorous comparison between say, DIBS general ledger package (\$400.00 for binaries) against PYRAMID'S general ledger package (\$2,000 installed by an OEM with help to the end user). Here, we are faced with conflict of interest on the part of OEM's whose business it is to sell such software to end users. There is a conflict between DEC and the OEMs, and between end users and the OEM's. The free flow vs. the withholding or channeling of information is crucial to all parties. It is going to be interesting to see how these conflicting areas are contained and managed.

Hardware configurations

VT100 against VT78
 LA120 against LA180 and LQP
 CPUs: 8a against micros against 8e, etc.
 RX02 high density floppies: reliability?
 RL01 against other hard discs
 Put 'em all together, they spell M-O-T-H-E-R !

Hardware/operating system interactions and limitations.

This gets a little confusing. Also, its hard to retain information provided by DEC or others as to what piece, particle, or gizmo will work with which version of what operating system. And, I don't need to tell you, it keeps changing. All this needs to be straightened out by DEC and kept current. It's just good salesmanship.

Systems analysis focused on the planning of systems based on the PDP8 that might need to change to PDP11 sometime in the future. How to keep from "painting oneself into a corner".

Potential impact within two years of the micro-11's, such as the PDP11/23, its use under WORD-11 or a future WPS-11, together with a discussion of the availability and advantages of the various operating systems, RSX-11, RSTS, etc.

All the above with a heavy emphasis on comparative costs going in!

I don't believe it is efficient to mix programmers and systems analysts with persons having little or no technical background. I do not wish to see a prohibition against end users commingling with experts, but I advocate the availability of two levels of instruction; high level for the high flyers and pragmatic fundamentals for end users.

Software exchange

To me, this is the least promising area in which to raise funds. Questions of title, liability, copyright, and value all come immediately to mind. However, I think the intent of the past should be protected and kept alive by those individuals who wish to interchange software on an ad hoc and more or less private basis. It is evident to me there are many high-minded, altruistic people working in this SIG who are willing to share their hard work openly and for little or no personal compensation. People like yourself Bob, and Johnathan Lockwood, Larry Eisenberg, and Jim Coryell. These people are, and will continue to be the backbone of the SIG.

If software interchange is to be rationalized, it will take something like what Gary Cole suggested: a printed listing of keywords enabling any of us to pick what he wants out of the mass of offerings.

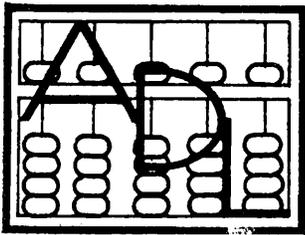
Best of luck and best wishes to all of you.

Yours very truly,

MUSSETTER REALTY, INC.

Robert Mussetter

cc: Gary Cole, DEC
Maynard, Mass 01754
Bill Lennon
Lawrence Livermore Radiation Laboratory
P.O. Box 808
Livermore, CA 94550



DATA SYSTEMS, INC.

6 Pinewood Drive
Monsey, New York 10952
November 28, 1979

(212) 737-3748

(914) 352-1698

Mr. Robert Hassinger, Chairman, 12 Bit SIG
DECUS
One Iron Way
Marlboro, Mass 01752

re: PDP-8 System

Dear Mr. Hassinger:

With respect to the above, several times we have approached the limit for the number of devices on this system. In searching for new devices, we have developed some device handlers. These include a combined paper tape reader punch handler (which I call PPT for paper tape). A handler that accepts form-feed characters for use on the new LA120 Printer. This handler also accepts the Control 'S' and 'Q' characters.

We also discovered an undocumented (and I believe it is even documented to the contrary) feature of OS/8 and OS/8 Build that allows the DSK handler to be specified to a device which has not otherwise been declared as Resident; i.e., on RK05's, SYS can be RKA0 and DSK can be RKBO without RKA0 and RKBO being installed in the System. Additionally, and this is a point totally undocumented, the DSK handler does not need to be Co-Resident with SYS. The only effect of this I have been able to discern is that Batch must run off SYS; it cannot run off DSK. This feature is very useful on RL01's, in which group of modules the RLOB Unit is not Co-Resident with SYS. I believe this is due to lack of space in the system handler. In order to help alleviate the above problems on the RK05 drives, we placed two SABR Programs on the San Francisco Symposium, a routine to read and write disks in SABR called, 'RWDISK'. This handler has the following calls and parameters. Call RDISK: (Unit, Block, Pages, Location) and the call to write it out. Call WDISK: Has the same parameters.

The first parameter Unit is defined as Two Units per drive (0/1, 2/3, 4/5, 6/7), the same as OS/8 uses. Each unit being 3,248 Blocks. A second program was put there by the name of 'IDKRD'. This routine has two parameters -- the Unit Number and Location. This routine is a function which returns a value of Zero if a disk is properly mounted in the drive and is ready. It also puts into the array location, one page of code that can be used as a disk header and label information. If the disk is not ready, it returns a negative value.

Very truly yours,

Aaron S. Weg
Exec. Vice President

November 28, 1979

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

#38 - PAGE 46

Dear Bob,

It may be of interest to some segments of the user community to learn that I have recently finished developing a RTS/8 version of the byte-mode floppy handler which was mentioned in issue #36 (pp 8-11). This version combines in one 3-page module all the code required to read a floppy in either 12-bit or packed byte-mode formats as well as offering a number of other special access modes which are available only to RTS/8 tasks. The code is one page larger than the standard DEC handler but the additional page is required anyway to improve the speed of 12-bit transfers on a VT/78 system. Unfortunately the power-fail code no longer fits within the body of the handler, so a fourth page would be required to include that option as well.

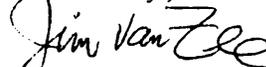
Because of the design of the OS/8 file-support task it was necessary to code the 'format-selection' bits as part of the 'unit' designation. Thus 'units 4,5' handle data written in a RT/11-compatible format, while 'units 6,7' use the special 'VT/78' format. 'Units 0-1', of course, use the standard 12-bit format while 'units 2-3' are currently undefined. I suppose that some enterprising soul could use those 'units' to handle COS-310 floppies at the expense of an additional page of code (or more). Such a module would then be able to handle a wide range of data formats.

The use of different 'units' for different formats is invisible to the OS/8 background. Thus from a user's point of view one still has 3 separate handlers: RXA0-1, D0-1 and F0-1, according to whatever devices one had originally installed in the system. The correspondence between 'D0' and 'unit 4' is established whenever one boots up RTS/8.

Performance has so far been tested primarily on a PDP-12 system with a 1.9 μ sec cycle time (and 5 μ sec IOTs!), and as expected, the throughput in 12-bit mode drops a bit in spite of the VT/78 enhancements. 'D' format suffers far more, however, because the time was already quite critical; but 'F' format runs at full speed. I anticipate that the performance on an 8/e will be much improved with all data transfers running at full speed in spite of the RTS/8 overhead. Hopefully this will also be true on 8/A systems.

From a system point of view, however, the slower transfer rate is not necessarily a problem because time spent waiting for disk rotation is not 'lost' as it is under OS/8, but is made available to other tasks which have something useful to do. It is rather exciting, for example, to watch a LINC-tape-to-floppy transfer: the tape never reverses and the floppy appears to be running at full speed, but in the meantime the null job keeps blinking the lights in the AC and the clock continues interrupting everything 1000 times a second!

Sincerely,



Jim van Zee
LAB DATA SYSTEMS
10320 Ravenna Avenue N.E.
Seattle, Washington 98125

P.S. I am enclosing a copy of the RK05 bootstrap which can be booted from either loc 30, or more conveniently, from loc 40. It uses Dick Murphy's scheme for preserving the date too, as well as allowing use of RKB0: as SYS:.



DECUS

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

CC 777

DEC-PERSONNEL - 126374

WR

DRP

E

1

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.