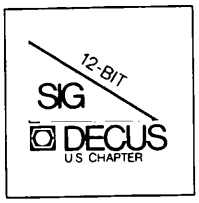

DECUS 12 BIT SPECIAL INTEREST GROUP
NEWSLETTER

Summer-Fall-Winter Number 40 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

Submissions are accepted at all times and are normally used in the next issue to go to press regardless of date of receipt. 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.

SIG UPDATE

No, you were not dropped from the mailing list. No, you did not miss the last three Newsletters. Yes, I have been having a lot of problems lately. Some of them involve the facilities I use to produce the Newsletter, some of them have to do with getting time to do the job and, most important, some of you who promised material for the Newsletter did not come through. Any one of these would not be a problem, but taken together they have resulted in a long delay between Newsletter numbers 39 and 40.

With luck, I will now be able to pick up the schedule again. You have to keep sending contributions though, or I cannot do it. If the Newsletter is still valuable to you, let me know and submit something.

All this comes at a time when we have received some good news from DECUS. The long anticipated need to start charging a fee for DECUS SIG newsletters has been put off for at least one more year. The DECUS/US Executive Board and the new DECUS Executive Director have been working with DEC to come up with SIG newsletter funding for the next year. I have gotten the feeling that the importance and value of the newsletters is much better understood now in those circles. It may well be that the funding can be continued even further, we shall see.

This seems to be a very important issue. I have been saying for several years that if DECUS imposes fees for the various SIG newsletters it will mean the end of many of them, quite possibly including our 12 Bit SIG Newsletter. It seems as though that would be a great loss for DECUS! If you have any thoughts on this, write to the US Executive Board at the DECUS address above and send me a copy if you like.

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

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

Special Steering Committee Advisor: Stan Rabinowitz

US_Symposia_Committee_Representative: Lee Nichols - see above

RTS-8 Working Group
Lee Nichols - see above

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

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

Symposium Software Exchange Committee

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

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

FALL DECUS SYMPOSIUM

The US Fall Symposium was held in San Diego the first week in November. For the first time in memory, there was no PDP-8 at the meeting for software exchange. This came as quite a suprise and disappointment to me and many of the attendees who consider software exchange to be one of the most important features of the Symposium program.

Gary Cole announced new DECstation 78 pricing (see separate item). Gary presented the new OS/8-OS/78 development and maintenance group who have spent the last few months gathering everything in the company about the OS/8 family including the entire SPR files. The new group seems dedicated to reversing the trend towards poorer and poorer software maintenance. Sometime soon we are supposed to hear about a new Digital Sofwate News program that everyone will be able to get for some "nominal" cost. A new release of OS/78 is being worked on and it is hoped it will be more than just a maintenance release.

As to the persistent feeling among some of our users that DEC has given up on the 12 bit machines, there were credible rumors circulating to the effect that a new product is in the final stages of development. The indications were that the decision on if and when to market something new are still under consideration and no commitments have been made as yet.

Anyone interested in these developments should watch the Spring Symposium very carefully. With a little luck, we may have a very interesting meeting next time.

NEW DECSTATION PRICES

At the San Diego Symposium Gary Cole announced new pricing for the DECstation 78 products. Gary indicated that the prices from the various sources within DEC (Catalog, Retail Stores, various Product Lines, etc.) will now be consistent. The new prices seem to be much more aggressive than in the past and seem very interesting. The following is what I could get on the fly during the session. There is no guarantee these figures are accurate, use them as a guide and check with DEC for more details.

Basic hardware prices:

VT78,RX02,LA34	$4995
VT78,RX02,LA180	$6895
VT78,RX02,LQP78	$7295

Software prices:

OS/78	$800
COS:	
Runtime only	$800
Unsupported system	$1600
WPS:	
without List Processing	$500
with List Processing	$900

(note: watch out for add-on cost for communications option!)

Typical System prices:

OS/78 system with LA-34	$5795
WPS with LQP	$7795
COS with LA-34	$6595

TU58 NOTES

In the last issue of the Newsletter, I mentioned that there is a problem interfacing and programming the TU58 for many of the 12 bit machines because you are supposed to be able to make the BREAK signal on the serial line interface you use to get to the TU58 and many 12 bit serial line interfaces cannot make BREAK. The same problem is an issue on other systems also. Jon Taylor from DEC's Microcomputer Products Group has provided me with the following information on this subject from Micronote number 86.

"The controller of the TU58 intelligent tape cartridge system can be connected to any serial interface that conforms to RS-422, RS-423 or RS-232C interface standards. This allows a TU58 drive to be connected to any non-DEC host computer that provides these interfaces. The TU-58 can be connected to such a host even though its interface is incapable of transmitting break ("space" condition) to the controller. This article explains the hazards involved with using such an interface.

"The TU58 DECtape User's Guide (EK-OTU58-UG) describes the radial serial protocol that is used by a processor to communicate with a TU58 controller. The protocol uses a break signal to initialize the controller in much the same way as the LSI-11 bus BINIT signal is used to initialize devices on the bus. Regardless of the state of the line protocol (and the controller), the controller will always detect the break signal as it is routed to the controller's "non-maskable interrupt".

"Break is normally used in two situations:

1. On power up, the TU58 continuously sends INIT bytes to the host. The host sends break and two INIT bytes. The TU58 responds with a CONTINUE byte and is ready for use.

2. If communications break down due to a protocol, line, or TU58 error, the host can restore order by sending a break and two INIT bytes. As above, the TU58 will respond with a CONTINUE and wait for further instructions.

"In situation one, a host without break capabilities would send just one INIT byte

and the TU58 will respond in the usual way with a CONTINUE. The host should be prepared to ignore one or two INIT bytes that may be seen before the CONTINUE byte (due to UART buffering).

"The absence of break in the second situation is the cause of less reliable operation.

"In most cases, the standard checksumming of messages and protocol handshaking will detect a protocol or line error and the state of the protocol will be known. To reset the TU58, the breakless host would send one INIT byte and wait for a CONTINUE byte response. However, in case an error occurs while the host is sending a packet to the TU58, more than one INIT must be sent, as the TU58 cannot distinguish the INIT from the packet it is expecting. There is a very slim chance in a write operation (one in 65536) that when the TU58 interprets two of the INIT's as a checksum word, the checksum will actually be correct and an erroneous write will occur.

"Very infrequently, communications may break down due to a TU58 controller malfunction (caused by power glitches or noisy environments). Without a break, the only way to reset the controller is to power it off and on.

"To avoid a possible malfunction because the controller cannot be initialized, Digital recommends that all serial interfaces to the TU58 be capable of generating a space condition."

I understand that the TU58 has already been interfaced to a PDP-10. Also, there was a very interesting paper given at the Spring Symposium on a related subject. It seems that the new release of RT-11 has handlers for the TU58 as a system device as well as non-system. The point of the paper was that instead of having a real TU58 at the end of the serial line the handler uses to interface to the tape, another computer was connected instead. A program that can simulate a TU58 runs in the remote computer and accesses disk storage to provide mass storage for the RT-11 system. In the case discussed in the paper, the program to simulate the TU58 ran as a foreground job in another RT-11 system. There is no fundamental reason why it could not be a task in an RSX-11M or RTS-8 system, however. This seems like an easy way to get remote, computer-to-computer communication. One simulator program could be used to service requests from any computer that had TU58 software, and, most interestingly, it gives a clean way to support very small, low cost configurations that do not have to have any mass storage of their own. In very small configurations, the usual floppy disks required for a minimal stand alone capability represent more than half the total system cost. With this approach, the software that runs in the small system is pure DEC, the only special software is a single TU58 controller simulator program or task for any particular operating system. Once this code is written for a particular operating system, it should serve all potential users.

NEWS FROM JIM VAN ZEE

"The last Newsletter (#39, March 1980) briefly discussed some considerations relating to the use of DEC's new 3M Cartridge tape drive, the TU-58(DECtape II), under OS/8. After talking to several people in the company, I gathered the impression that DEC wasn't planning to offer an OS/8 handler for this device, so I thought I would report on my efforts to develop one.

"The TU58 is a rather attractive mass-storage device from a number of points of view: (1) it is available as an 'off the shelf' OEM product from electronics supply houses (so you don't even have to deal with Digital to get one!) (2) the cost is quite low - about 1/3 the cost of a floppy disk system; (3) a single tape can store over 1/4 MB of data - just a little bit more than a single-density floppy can store using my RT-11 compatible byte-mode handler; (4) the data is stored in a numbered block-structured format, which allows random access operations, as opposed to a 'record gap' format; and (5) interfacing is accomplished via a standard RS232 'serial line' port at speeds up to 38.4 KB. Because of the 'universal nature' of this type of interface, the TU58 can be easily attached to many different kinds of processors, allowing files to be easily moved from one machine to another. It is thus important to write data on the tape in a manner compatible with existing file conventions for this device (i.e. a RT-11 file structure).

"The handler I have written supports the TU58 as a 'non-system' device under OS/8, and can be assembled to work on -any- PDP8 processor. It adds the devices 'DTU0' and 'DTU1' ('DEC-Tape Units 0,1') to the system, each of which is 682 OS/8 blocks long (675 'free' blocks); this uses all but two of the 2048 pre-formatted records on the tape - the last two records cannot be accessed under OS/8. ASCII files written with this handler are readable on a PDP-11 (and vice-versa), however the directory structure is different for the two operating systems, so one must use a little program to decipher the directory area if access by filename (rather than by location) is desired. It is impossible to write a -system- handler for the TU58 unless one can do the checksum calculation in hardware or in some hidden ROM code; I have developed versions for both cases, as well as a modification for the KL8E interface which allows one to use the TU58 as an OS/8 system device.

"The use of 'messages' to control the tape drive, instead of simple machine instructions, nearly made it impossible to code a handler which would fit in two relocatable pages. Indeed, the initial outline of the code came to over 300 instructions, not counting relocation or cross-page linkages! To shrink this down by some 70 instructions it was necessary to apply every 'ounce' of ingenuity available, as well as to make a number of compromises, which, fortunately, do not affect normal operation. For example, there is almost no protocol checking: when a status message is sent by the TU58, the handler will wait for it, but, for the most part it does not check any of the message elements except the 'success code'.

"Similarly, there is no checksum or byte-count verification when reading data. This is not a very serious omission since the TU58 does its own error checking, so the only thing the handler could verify is the integrity of the connection from the controller to the interface and the functioning of the receiver. These elements are less suspect than the tape itself, so this deficiency in the handler has little effect on expected performance. Another feature which had to be left out was the usual 3 'retries' after an error; this again is not really necessary since the TU58 automatically retries an operation -8- times before signaling an error condition anyway.

"It was also necessary to make a tradeoff between including the normal 'CTRL/C' check and a second entry point. The tape controller will handle two drives, so it is attractive to have a single handler which can access either one. Without the hardware checksum modification mentioned earlier, however, there is simply not enough room for the keyboard check unless one uses a separate handler for each drive. Because of the limited number of 'handler slots' in the OS/8 system, I have generally preferred to

live with the inconvenience of waiting for the calling program to respond to a CTRL/C rather than having two separate handlers. Unfortunately, not all CUSPS have this check in spite of the fact that it is a system requirement (due to the lack of a keyboard check in all system handlers). CCL, alas, is one of the offenders, which can mean unfortunate delays when you are using the TU58 as a system device.

"As mentioned in the previous Newsletter, the TU58 has one requirement which is not a standard feature of PDP8 serial line interfaces: it needs to be sent a 'break' after it has been turned on before it will listen to anything else! This signal is also used as a 'master reset' to interrupt the message protocol, for instance to initiate a 'boot' operation. Since none of the standard interfaces have a specific mechanism for generating a break it was necessary to resort to a bit of 'trickery'.

"The first idea that came to mind was to simply send a stream of nulls without waiting for the 'done' flag. When I tried this on a PDP-12 using a 'M707' interface (this is the standard SL interface card for all 'pre-Omnibus' machines) it didn't work. A study of the circuitry (and a little work with a 'scope) disclosed that the difficulty lay in the method used to load the shift register: instead of a 'jam' transfer as one might have expected, the 'TLS' instruction actually just 'ORed' the data into the buffer. This meant that the 'guard' bit never got cleared which prematurely terminated the 'break'. Adding a single wire to clear the low-order flip-flop solved the problem without affecting other uses of the same interface.

"Many different serial line interfaces are available for 'Omnibus' machines: DEC initially sold the 'KL8E' (M8650), followed later by the 'KL8-JA' and then the 'KL8A'; in addition, several other companies have produced similar interfaces. Unfortunately all of these except the KL8E use a UART chip to simplify the design, with the result that they cannot send a break without substantial modification. The KL8E, on the other hand, uses a design somewhat similar to the M707 card, and can thus be made to send breaks without any modification whatsoever. Although this module was originally sold as a '1200-baud' interface, it in fact works quite well at 9600 baud (or higher!). Several used equipment dealers (such as Omni Systems in West Caldwell, New Jersey: (201) 335-6919) have these cards at quite attractive prices, which makes them the ideal interface for this application.

"Owners of VT/78 DECstations (or the equivalent WS/78 Word Processor), can simply plug the TU58 into the 19.2KB 'SLU3' port on the rear panel. In this case the break signal is created by momentarily changing the baud rate via the 'TSB' instruction. Differences between various interfaces such as this, as well as the choice of device codes, have naturally been placed in conditional coding, so a handler which will work on a given machine can be quickly generated by simply defining a few symbols such as 'KL8E', 'M707', 'VT78', 'RECDVC', 'XMTDVC' and the like. The handler does not use any instructions which cannot be executed on all PDP8 (or PDP12) processors; the penalty for this was only one or two locations, but the result is that the handler is just as universal as the TU58 itself!

"In fact, the development of this handler has opened the door to a new generation of low-cost PDP8 systems! Since the connection between the TU58 and the CPU is nothing more than a simple communications line, it is immediately obvious that the TU58 itself can be replaced by an emulator program running on a large time-sharing system. This leads to a simple implementation of multi-processing systems having a mixture of 8s, 11s and VAXs. To test this idea, Jim Gladden at the University of Washington wrote a simple TU58 emulator program for the VAX which makes a user's port

behave just as though it were a 'real' TU58 tape drive. When this program is running one can actually boot up an OS/8 system (ON THE VAX!), using the communications line as the only mass-storage device! In times when 'PDP8 chips' only cost $10, and a complete 32K machine can be had for $1-2K, this solution to the high cost of mass-storage devices is obviously exceedingly attractive for a number of data acquisition and control applications.

"What about speed? The TU58 is advertised as a -FAST- tape drive with "an average access time of 9.3 seconds". Indeed DEC's marketing department seems to feel that this is one of the most significant features of the tape drive. Since it ACTUALLY TAKES about 33 seconds to wind the tape from one end to the other, one might have expected the 'average access time' to be a bit closer to 16 seconds - so where does DEC get the extra factor of two? The answer is that tape blocks are interleaved, so at any given position on the tape one is equally near 4 completely different areas. For example, blocks 1, 171, 341 and 511 are all near each other. To go from the directory in block 1 to a file at block 350, for example, only takes a fraction of a second.

"But this scheme isn't as marvelous as it looks on paper! In actual fact, an operating system such as OS/8 does not access file-stuctured mass storage devices in a 'random' fashion, and in particular, the expected use of the TU58 as a medium for file interchange implies exactly the opposite: namely that moderately long files (up to 675 blocks at a time) will be copied onto such tapes in order to be transferred to a different machine. For applications such as this the 4-way interleave is a SERIOUS DEFICIENCY because the tape must be wound back and forth 4 times (a total of 4 minutes of pure tape motion) simply to pass over all the blocks. A regular DECtape or LINCtape on the other hand has about the same 'end-to-end' time, but is 4 times faster than the TU58 simply because the blocks are not interleaved!

"Thus my initial response to TU58-based OS/8 systems was that they were hopelessly slow! It took 45 seconds to get a response to a '.DA' command, over a minute to get a directory listing, and about -5- minutes to complete a '.RES/E' command! However, in talking to several users of PDT-11 systems which use the same tape drive, I found that they felt such response was acceptable for the intended applications. Furthermore, my enthusiasm picked up again after we got the VAX emulator working, since the speed was now determined only by the serial line. At 9600 baud, for example, a .DA command only took 10 seconds, a directory listing about 15 and a resource check was completed in about 25. These times are almost identical to those observed on DEC- or LINCtape systems, and could be cut in half by simply going to a faster communications line. Such systems are obviously slower than floppy disks, but when used with languages such as U/W-FOCAL (which 'grew up' in a DECtape atmosphere and is therefore optimized for slow system devices), the response is really not bad! FOCAL programs, for example, can be loaded in just a second or two (from the VAX), which keeps the system at the desired interactive level. I certainly wouldn't recommend trying to develop a FORTRAN program on such a system, however, unless you never make mistakes!

"Two final comments: the bootstrap which I devised is (in my view) just as remarkable as the handler itself! Unlike other routines of this genre, the TU58 bootstrap must deal with the problem that different machines are likely to have serial line interfaces with different device codes. To allow TU58 system tapes to be interchanged between different installations, it was necessary to copy the IOTs used in the primary boot into the handler itself. Hence by merely changing these

instructions one can adapt the system to any sort of interface without changing a single bit in the handler. The primary boot is only 30 locations long (see the attached listing), and can be easily put into a ROM if desired. It has an option for booting up a system on either drive, although this is not done automatically as it is in the floppy disk bootstrap. The other notable feature is that the OS/8 date is preserved during a boot - something I think all bootstraps should do! (Incidentally, I have a patch for the RK08 and RK8E bootstraps to preserve the date, should anyone be interested...)

"Of course it is necessary to modify the interface in order to use the TU58 as a -system- device simply because there isn't enough room to perform the checksum calculation in the space allocated for the system handler. As I mentioned earlier I have designed and built a little addition to the KL8E board which does this using only 4 ICs - one of which, however is a microprocessor! After considering a number of circuit designs it became obvious that this was the most efficient way to do the checksum, and I would like to recommend the Intel 8748 to others looking for a simple 'interface' processor. It is not very good as a general-purpose CPU, but it is excellent for interface applications. To those acquainted with the LINC or the PDP12 the 8748 may seem somewhat familiar! The 'page' and 'field' boundaries, so familiar to '8 programmers, are all there too! But never mind that it isnt a 'Z80'! - for applications such as BCD-to-binary conversion (i.e. read the number, convert it to binary and give it to the '8), the 8748 is just about perfect. This can be done with 2 ICs and about 50 bytes of code!

"Of course interfaces using microprocessors have software bugs in addition to hardware bugs, and as an example of such difficulties, it appears that there may be a 'bug' in the code used by the 8085 which runs the TU58! The unit I was testing would not do 'high-reliability' operations when it was accessed in 'special addressing' mode. This was not a 'known bug' when I reported it to the people in Maynard, but in spite of the potential for having thousands of defective systems in the field, I wasn't able to arouse much interest in the problem. (About a week later, however, DEC did raise the price of the TU58 by 22%, perhaps to cover the cost of an anticipated recall??) It will be interesting to see what sort of 'SPR' service is developed for ROM code!

"Finally I would like to mention several suppliers (in addition to DEC) who sell packaged TU58 systems: 1) Hamilton Avnet Electronics (offices in many major cities) sells both the 'raw' tape drive as well as the recently announced packaged systems; 2) G C Controls in Greene, N.Y., (607) 656-4117 has a packaged system, as does 3) General Digital Corporation in Huntsville Alabama (205) 883-1700. There may be others; if so I hope they will call attention to themselves.

- - - - - - - - - - - - - - - - - -

"I also recently completed another amazing handler: this one is a system handler for the RX02 (DSD-440) floppy disk which supports BOTH drives, rather than only one. The advantages are perhaps somewhat modest: faster access to the second drive (since the handler is always resident), and two more pages of memory for programs which are smart enough to use space not used by handlers. Another benefit is that the OS/8 date is preserved during a 'boot' - a feature I deem incumbent upon all system handlers since so many users seem to forget to enter the date after they boot up the system. The biggest advantage, however, is that this handler frees up one of the 8 'handler slots' in the system area, which allows you to include an entirely new device in the

system.

"The only real difficulty I encountered was not with the handler itself but with the RXCOPY program. RXCOPY, although it calls the command decoder to find out what to do, does NOT use the handlers you specify, but rather infers from the value of the handler entry point returned by the C.D. what it THINKS you told it (i.e. to copy from SYS to RXA1, or whatever); it then uses its own internal handlers for the actual data transfer so that it can make a true image copy of the disk. (The OS/8 handlers use 12-bit mode, so 25% of the disk is normally not accessible.) Such inferences can obviously lead to trouble if the entry points of a user-written handler do not agree with those used by DEC, and indeed when I first tried to use the .DUPLICATE command I managed to copy a brand new BLANK disk onto my development system!

"This disaster led to an immediate patch to the RXCOPY program so that it would behave itself (either with my improved handler, or with the standard DEC handlers), after which I have been completely happy with the new system. Following the DEC convention, the disk can be booted up on either drive, that drive automatically becoming 'RXA0' as well as 'SYS'. But to keep the innocent floppy flipper happy, the handler automatically assigns the name 'RXA1' to the -other drive- (whichever one it may be physically), so user programs work properly no matter how the system is booted up! Incidentally, the same floppy disk can be booted up on -any- 'DEC-compatible' floppy disk system (RX01, RX02, DSD-210, 440, 480, etc.), provided that the hardware can read the disk. Thus a single-density floppy with this handler can be used on either the RX01 or the RX02. In general the handler will automatically adjust itself for use with whatever media is inserted in the drive being referenced.

"The last piece of news from the land of the smoking volcano is that I have developed a little overlay for the current version of U/W-FOCAL (V4G) which makes a rather dramatic improvement to the symbol storage routine. All versions of FOCAL up to this point have used the original 'sequential access' method of storage in which variables are located by checking each one (starting from the first) for a matching name and subscript. This is a very simple method to implement which works quite well in the 8K and 12K versions because of the limited number of variables available (less than 200). In the 16K version, however, things get a little slow when you need to reference the 500th variable, especially if you should be so unlucky as to define a loop index AFTER you have filled a huge array. In that case FOCAL can spend most of its time locating the same index variable over and over again instead of doing useful calculations!

"The improved routine uses a 'hash-coding' method to calculate the approximate storage location from the name and the subscript. It then tries a 'direct lookup', searching nearby locations if the variable is not there. This scatters the variables throughout memory and thus complicates things like the ZERO command and the 'secret variables', but the improvement in access time is truly dramatic. Furthermore, since speed was no longer an issue, I wrote the routine to use all available memory (up to 5 fields) for the symbol table. This is controlled by the system .MEM command as well as by an internal switch setting, and can be dynamically changed during a program by issuing an appropriate 'ZERO' command. The maximum number of variables is now 3405, compared to the previous limit of 676.

"This gives UWF essentially the same calculational power as DEC's FORTRAN IV in terms of array sizes, but for large calculations, UWF has the advantage of higher accuracy ('10-digit' instead of '6-digit'), which is a very telling consideration when

performing operations on large matrices. For example, I once tried to invert a little 8x8 matrix using FORTRAN and got errors of 300% in the final result! When the same routine was converted to U/W-FOCAL, the errors were only in the 5th decimal place! I notice that almost all of the 'micro' BASICs, for example, have at least 8 digit accuracy (as do the $4 pocket calculators!), while many have 10 digits or even more. I realise that the new Commerical BASIC has 'string arithmetic' operations which give 15-place precision, but that feature does not meet the needs of the PDP-8 based scientific and research community. While buying a Floating Point Processor so you can use double precision in FORTRAN is also a solution to this problem, those with limited budgets may discover that UWF now provides sufficient variables and adequate accuracy for most 'reasonable' problems.

Jim van Zee - Lab Data Systems - 10320 Ravenna Ave. NE - Seattle, WA. 98125"

(Original) 27 June 1980
(Revised) 8 October 1980

```
/10  OS/8 BOOTSTRAP FOR THE TU58          PAL8-V12B 13-JUL-80 PAGE 3

             / THE PRIMARY BOOT USES ONLY 30. INSTRUCTIONS!  IT CAN
             / BE EASILY CHANGED FOR USE WITH DIFFERENT INTERFACES.
             / THE IOTS ARE COPIED INTO THE SYSTEM HANDLER AFTER IT
             / IS LOADED, MAKING IT 'MACHINE-INDEPENDENT'!

             / DO NOT CHANGE THE LOCATION OF ANYTHING FROM 32-53!!!

       0020          *20              /VERSION FOR 'UARTS' (VT78)
00020  6333  BSTART, TTSB             /SET TO BAUD RATE TO 50
00021  4050          JMS WR1          /SEND A LONG NULL
00022  1034          TAD BYT2         /LOOKS LIKE '17'
00023  6333          TTSB             /RESTORE 19200 BAUD

       0020          *20              /VERSION FOR 'KL8E'S (M707)
00020  6336  BSTART, TTLS             /RELOAD OUTPUT BUFFER
00021  2050          ISZ WR1          /OVER AND OVER AGAIN
00022  5020          JMP .-2
00023  4050          JMS WR1          /END WITH A FULL CHAR.

00024  7307          CLA CLL IAC RTL  /MAKE A '4'
00025  4050          JMS WR1          /SEND 'INIT'
00026  7004          RAL              /GET A '10'
00027  4050          JMS WR1          /SEND 'BOOT'
00030  7200          CLA     /IAC     /TO BOOT UP ON UNIT 1
00031  4050          JMS WR1          /SEND <0> FOR DRIVE 0

00032  7010  SLIM,    RAR              /FALL INTO SLIM LOADER
00033  7012  BYT1,    RTR
00034  3177  BYT2,    DCA 177          /'DCA 200' = 'DCA 0' !
00035  2034  BYT3,    ISZ .-1
00036  7410  B7410,   SKP              /USED MANY DIFF. WAYS!

00037  0032  RD1,     SLIM             /ENTRY POINT OF READ SUB
00040  6321  KSFC,    KKSF
```

```
00041  5040          JMP .-1          /WR1 CLEARS FLAG INITIALLY
00042  6326  KRBC,   KKRB
00043  5437          JMP I RD1        /CHAR IN AC<4-11>, <0-3>=0

              / HERE IS A LIST OF ALL SPECIAL IOTS REQUIRED:

00044  0000  IOT1,   0   / A '0'      /CLEAR THE CHECKSUM
00045  0000  IOT2,   0   / MEANS      /ADD TO THE CHECKSUM
00046  0000  IOT3,   0   / DON'T      /GET LOW ORDER C.S.
00047  0000  IOT4,   0   / PATCH      /GET HIGH ORDER C.S.

00050  0000  WR1,    0                /WRITE SUB
00051  6336  TLSC,   TTLS
00052  6030          KCF              /CLEAR INPUT FLAG
00053  6331  TSFC,   TTSF
00054  5053          JMP .-1          /WAIT FOR 'DONE'
00055  5450          JMP I WR1

              / ABOVE CAN USE 'DCA 177' IN PLACE OF 'KCF', WITH THE
              / ADDITION OF 'KCC; TAD 177' AHEAD OF THE 'JMP I WR1'
```

NEW DECUS LIBRARY SUBMISSION

OS/8 VAX Handler - from Jim Van Zee, Laboratory Data Systems, Seattle, Washington. "The VAX handler allows any PDP-8 or PDP-12 to send files to (or receive files from) a VAX computer. No special programs are required to do this: the handler may be used with any OS/8 program and presumes only the existence of the $COPY command on the VAX. Only ASCII data may be transferred, but commands as well as data may be sent to the VAX, hence an OS/8 program can use this handler to do 'parallel processing' on data collected by the '8 or '12. Restriction: OS/8 FORTRAN-IV programs using this handler for input can only put output files on the system device."

Jim notes that this handler should be usable with OS/8, OS/78, DECsystem/8, PS/8, and OS/12 (since they are all really the same system, at least as far as handlers are concerned). The documentation is on the magnetic media and the code is written in PAL8. A serial line interface is required for the connection to the VAX. The available media/service charge codes are: Write-up and Listing (DA), DECtape (HA), Floppy Diskette (KA).

PASCAL NOTE

I noticed the following note in the Structured Languages SIG Newsletter (Vol 4 No 2 - Sept 80):

> "PASCAL-S for PDP-8: Prof Stegbauer from the HTL Modling (near Vienna) has implemented a compiler interpreter of Wirth's Pascal subset PASCAL-S on a 28k PDP-8 for teaching purposes. The operating system is OS/8. All messages are in German. You can receive a free copy if you send me an OS/8 formatted DECtape."

The newsletter format is a little hard to interpret, but I assume the note was from DI. K. Mayer, Institut fur Physik, Osterreichische, Studiengesellschaft fur Atomenergie Ges.m.b.H., Lenaugasse 10, A-1082 WIEN, Austria. Any inquiries should be

directed to him. If you find out anything interesting, please let me know for this newsletter.

WPS <-> OS/8 FILE CONVERSION SOFTWARE

I recently received the following note from Kenneth L. Thompson:

"This letter is in response to your call for information concerning file conversion programs as printed in January 1980, No. 38, Page 4.

"1. Since August 1978 we have been using a word processing (WPS-8) to OS/8 ASCII conversion program that we produced. Our procedure is:

a. Convert all word processing characters to an OS/8 file.
b. Use a FORTRAN based program to edit out various word processing based characters. (One probably could do a faster job here with TECO)
c. We use a TECO operation to un-string and produce an EDIT compatible line image file. (not required if file is used for input to FORTRAN program which is our normal mode)

"This WPS to OS/8 conversion program runs on OS/8 and OS/78. As I said, it has been used extensively by us and our clients since August 1978. Hence, it appears to be bug free.

"2. Recently one of our clients requested that we write a conversion from OS/8 ASCII file to word processing (WPS-8). We are currently debugging this code. The status therefore is "not for sure" but "maybe".

"3. At the San Diego DECUS 1979 a DEC representative verbally expressed an interest in a business relationship to acquire rights to our conversion programs. We are waiting until we know for sure about our success with the OS/8 to WPS conversion before pursuing a possible business relationship with DEC."

Mr. Thompson is president of American Systems, Inc., 9875 Looking Glassbrook Road, Grand Ledge, Michigan 48837 - phone (517) 626-6301.

FROM EARL ELLIS

"Effective IMMEDIATELY, I fear that I must withdraw my name from several 12-Bit SIG activities. On April 1, 1980, I will leave the PDP-8/E for a position using PDP 1103's, 1123's, and 1134. For the moment, at least, I'm not going to be as active is DECUS as I have in the past. I'm going to have to get up on RT-11 and RSX11M and I expect this to be 'all consuming'.

"Please remove my name from the Symposium Software Exchange Committee. Any one wishing to communicate with me after 1 April should contact me at the address below.

"The 'Virtual Eight Users Group', the '9-Track for the PDP-8', the 'FOCAL SIG', and the 'ETOS Users SIG' will all have to find new people to run them. Of these, only the 'FOCAL SIG' realy crosses machine lines, I have names for implementors/users on PDP-8, PDP-11, IBM-370, NOVA 1200, UNIVAC 1108, Z80, 6502, 6800, and others. FOCAL IS NOT DIEING. I will hold these names until someone steps forward.

"There have been several meetings of the 'Virtual Eight' groups, mainly at Symposia. ETOS, MULTOS, and MULTI-8 are the principal supported packages to date. Several other attempts have been made, including 'DUALOS' (using RTS8). Since a big PDP8 (32K, 50mB Disk, and ETOS) system that supports multiple COS &/or OS/8 (up to 17 users) is available for less than $30K (no Terminals) I think use of Virtual Eights will increase. MULTI8 is heavily used in Europe, ETOS has about 200 users. Among problems of these systems is that they each require at least one person who is knowledgeable enough to be a 'System Manager'.

"I've uncovered a bug in TECO which someone needs to examine. This applies to the DECUS version and the version I've been sending out. If it is assembled with MACREL V2, and V2C. The ERROR is at line 1107, ID-REDEFINED TAG 'SCOPY'. It seems that SCOPY is defined as a GLOBAL = 1514. It appears that this is due to code that's conditionalized out during PASS1 then assembled during PASS2 or something like that so that the address of SCOPY is shifted. This has been reported to me by Jim van Zee (My version) and GENRAD (DECUS version) It seems to me there may be a MACREL Version problem here, and if true, should we distribute MACREL.SV with the TECO source so that it can be assembled? GENRAD tried all versons of MACREL they had, including the latest Factory version and wound up looking at the error listing, FUTIL'ing in the patches.

"I hope that this is not Good Bye to the PDP8 and all the 12-Bitters I've met over the years. I may be able to convince my new boss that Symposia are of value, and RT-11 has always had close ties to OS/8. For the moment, at least, I plan to stay in touch through Jim van Zee.

"My new mailing address is Earl T. Ellis Jr., Software Project Manager, DOUGLAS RANDALL Division of Walter KIDDIE Inc., 6 Pawcatuck Avenue, Pawcatuck CT, 02891, (203) 599-1750"

HUTTON + ROSTRON ARRAY PACKAGE FOR USE WITH PAL8

Lars Palmer forwarded the following article:

INTRODUCTION

One of the problems of programming in a low-level language is the difficulty of handling arrays, especially those of more than one dimension. The Hutton + Rostron Array Package provides simple array-handling procedures for use with PAL8 in a configuration with at least 8k storage. The arrays may be 1- or 2-dimensional and are named with a single ASCII character. The lower bound of all arrays is 1. A maximum of 12 decimal arrays may be called in any program, but this may be increased by modifying the package.

Use of the package required 52 octal locations in the user's program. The subroutines occupy locations 7200-7777 in the highest available field. Array storage is from 7177 in the highest available field backwards to 0000 in the lowest unused field.

Parameters HIGFLD (highest available field) and LOWFLD (lowest unused field) must be set by the user. These settings determine the maximum number of cells available for storage:

HIGFLD-LOWFLD	maximum number of array cells available (decimal)
0	3712
1	7808
2	11904
3	16000
4	20096
5	24192
6	28288

USE OF THE PACKAGE

Before using any of the subroutines, the user's program must make the call:

JMS ARRAY

This sets up the initial parameters. After this call, three subroutines are available for use:

1 JMS SETA
"A
B
C

where:
"A (note double quotes) is the array name (one ASCII character)
B is the limit of the first dimension
C is the limit of the second dimension (C=0 for 1-dimensional arrays)

This reserves space for the named array and sets the array to zero.

Examples:

```
JMS SETA  /sets K[1:12,1:34] to zero
"K
12
34

JMS SETA  /sets M[1:12] to zero
"M
12
0
```

2 JMS PUTA
"P /array name
A

```
   B

   This deposits the accumulator in array location P[A,B]

   Example:

   TAD (1234 /P[6,72]:=1234
   JMS PUTA
   "P
   6
   72


3  JMS GETA
   "Z                /array name
   A
   B

   This sets the accumulator to the value of Z[A,B]

   Example:

   JMS GETA  /AC:=Z[12]
   "Z
   12
   0         /0 for 1-dimensional arrays


ERROR MESSAGES

There are three error messages:

ARRAY STORE FULL  :  more than 12 decimal arrays called
                     available fields for storage are full

ARRAY SUB OFLO    :  calling bounds exceeded

ARRAY ABSENT      :  call of array without call of JMS SETA

After any of these error messages, control returns to the keyboard
monitor


METHOD

If n is the highest available field, locations n7704 to n7777 contain
the array directory for 12 decimal arrays (ie 12 decimal * 5 words):

WORD 1  -301   /negative of array name "A
WORD 2  12     /bound 1 = 12
WORD 3  34     /bound 2 = 34
WORD 4  30     /start field*10 for first cell
WORD 5  5432   /start location for first cell
```

Array storage starts at n7703 and works backwards as follows:

location	A[1:4]	B[1:4,1:3]
n7703	4	4,3
n7702	3	3,3
n7701	2	2,3
n7700	1	1,3
n7677		4,2
n7676		3,2
n7675		2,2
n7674		1,2
n7673		4,1
n7672		3,1
n7671		2,1
n7670		1,1

An example program listing was enclosed but is too long to include here. For more a copy contact Lars, or, if that does not work, then me [RH].

ENCODE/DECODE FOR FORTRAN IV

Lars Palmer forwarded the following routine that can give FORTRAN IV the functionality of the ENCODE and DECODE statements.

```
/ RALF SUBROUTINE "CORE" TO IMPLEMENT ENCODE/DECODE
/ A.WINDRAM  GRI  14-MAY-80, 29-AUG-80
/       CALL CORE(N)    TO IMPLEMENT CORE DEVICE
/ THE CORE ROUTINE THEN CO-OPTS FORTRAN STREAM N        0<N<10
/
/ A RECORD MAY BE WRITTEN TO STREAM N, AND MAY THEN BE
/ READ FROM STREAM N AS MANY TIMES AS REQUIRED
/
/ RESTRICTIONS:-
/ 1)    STREAM N MUST BE UNUSED (ELSE THE CALL IS NO-OPED)
/ 2)    RECORDS LONGER THAN 133 CHARS WILL LOOP BACK &
/       OVERWRITE FROM THE BEGINNING OF THE RECORD
/ 3)    USES FRTS LOCATIONS 00125-00130

        FIELD1 CORE
        JA      START   / JUMP ROUND BASE PAGE ETC
        TEXT    +CORE  +
BPAGE,  F       0.0
/ *** BEWARE OF AIX REGS HERE (MAYBE) ***
XPAGE,  0;0             / INDEX REGS 0 & 1

/ FILL UP BASE PAGE WITH CONSTANTS, VARIABLES ETC
CM212,  -212
CM3,    212-215
C215,   215
C377,   377
```

```
CM10,   -12
C9,     11

PATCH=125               / FRTS PATCHED AT 125-130
DSRN=4233               / DSRN TABLE STARTS HERE WITH UNIT 0

ADSRN,  DSRN            / ADDR OF DSRN TABLE
ADSRN0, 0               / PTR TO DSRN ENTRY
APATCH, PATCH           / ADDR OF PATCH TO FRTS
CORPTR, ADDR    BUF     / POINTER & ADDR OF BUFFER
CORSAV, ADDR    BUF2    / TEMP & END OF BUFFER INDICATOR

/ START HERE TO SAVE SPACE
START,  STARTD
        JA      START2  / JUMP TO NEXT BIT

/ FIX BP VARIABLES 10 & 11 FOR TRACEBACK & EXIT
        ORG     BPAGE+30
        FNOP
        JA      BPAGE   / FOR TRACEBACK
        FNOP
GOBAK,  JA      .       / RETURN JUMP (FOR TRACEBACK)

/ NOW REAL START CODE
        BASE    0
START2, FLDA    30      / GET RETURN ADDR
        FSTA    GOBAK
        FLDA    0       / GET ADDR OF PAR LIST
        SETB    BPAGE
        SETX    XPAGE
        BASE    BPAGE   / SWAP TO OUR BASE & INDEX
        FSTA    BPAGE
        LDX     1,1
        FLDA%   BPAGE,1 / GET ADDR OF 1ST PARAMETER (N)
        FSTA    BPAGE
        STARTF
        FLDA%   BPAGE   / GET VALUE OF N
        ATX     1       / FIX IT
        TRAP4   CORE8   / CALL REAL ROUTINE
        JA      GOBAK   / & RETURN TO CALLER

CORE8,  0
        JMP BUF+1       / DO ONCE-ONLY CODE (PATCHED TO CDF)
        TAD XPAGE+1     / GET VALUE OF N
        TAD CM10
        CLL
        TAD C9
        SNL CLA
         JMP COR8X      / NOT 1-9 !!!
        TAD XPAGE+1
        CLL RTL
        RAL
        TAD XPAGE+1     / N * 9
```

```
        TAD ADSRN
        DCA ADSRN0      / GET ADDR OF APPROPRIATE DSRN TABLE
        TAD% ADSRN0     / STREAM IN USE ?
        SZA CLA
        JMP COR8X       / YES ; LEAVE WELL ALONE
        TAD APATCH
        DCA% ADSRN0     / MAKE NEW INTERNAL HANDLER
        ISZ ADSRN0
        CLL CML RTL     / 2
        DCA% ADSRN0     / INHIBIT FORMS CONTROL
        TAD CORPTR+1
        DCA CORPTR      / PTR TO BEGINNING OF BUFFER

COR8X,  CIF 0
        JMP% CORE8

/ ACTUAL CODE TO DO CORE WORK
CORE2,  DCA CORSAV      / SAVE AC
        TAD% APATCH
        DCA CORE8       / SAVE EXIT ADDRESS
        CDF 10
        TAD CORSAV
        SNA
        JMP CORIN       / INPUT OPERATION
        AND C377
        TAD CM212
        SNA
        JMP CORX        / IGNORE LF
        TAD CM3
        SNA
        JMP CORCR1      / GOT CR ON OUTPUT
        TAD C215
        DCA% CORPTR
        ISZ CORPTR
        NOP
        TAD CORPTR
        TAD CORSAV+1
        SZA CLA
        JMP CORX        / NOT OVER BUFFER END
CORRBP, TAD CORPTR+1
        DCA CORPTR      / RESET PTR TO BUFFER START
CORX,   CDF CIF 0
        JMP% CORE8

/ CR SEEN ; END BUFFER
CORCR1, DCA% CORPTR
        JMP CORRBP

/ INPUT ; GET NEXT CHAR
CORIN,  TAD CORPTR
        TAD CORSAV+1
        SNA CLA
        JMP CORCR2      / CR IF AT END
```

```
        TAD% CORPTR
        ISZ CORPTR
        NOP
        SZA
        JMP CORX

/ CR ON INPUT
CORCR2, TAD CORPTR+1
        DCA CORPTR
        TAD C215
        JMP CORX

/ BUFFER
BUF,    0               / BUFFER GOES HERE
/ START OF ONE-TIME CODE
        JMS% CORSAV+1   / CALL ANOTHER ROUTINE
        DCA CORE8+1     / KILL 1-TIME CODE
        JMP CORE8+1     / & CARRY ON

        ORG     .+177&7600      / MOVE TO NEXT PAGE

BUF2,   0               / START OF NEXT PAGE
        JMS BUF3        /*** MUST BE AT 2ND LOC ON PAGE ***

/ INIT CONSTANTS
ACS,    ADDR CORSAV
C176,   176
AL1,    PATCH+1
        PATCH+2
        PATCH+3
PI1,    CIF 10
        JMPZ% PATCH+3
        ADDR CORE2

BUF3,   0
        ISZ ACS+1       / ADJUST ADDR TO CORSAV+1
        CDF 10          / JUST IN CASE
        TAD BUF3
        TAD C176        / 1ST LOC OF NEXT PAGE
        CIA
        DCA% ACS+1      / MAKE BUFFER END MARK
CDF0,   CDF 0
        TAD PI1         / APPLY PATCH TO FRTS
        DCA% AL1
        TAD PI1+1
        DCA% AL1+1
        TAD PI1+3       / ADDR TAKES 2 WORDS !
        DCA% AL1+2
        TAD CDF0
        JMP% BUF2       / RETURN
        END
```

HELP - TENNICOMP TP1371

Can anyone help professor Brody with his tape:

"Desperately need someone to repair Tennicomp Systems MiniDec (TP1317). Call collect to Prof. S. S. Brody, (212) 598-3135."

The address is Department of Biology, New York University, 952 Brown Building, Washington Square, New York, N. Y. 10003.

HELP - BATCH BUG

Friedemann Brauer sent the following question:

"I'm having trouble with OS/8 V3D BATCH. After an output error F4 compiling stops (correct), the load module cannot be generated and the BATCH job ends (correct). But then DSK:=SYS: was set, which we use for two different devices !!! Only rebooting the System cured the problem."

Can anyone explain? Suggestion: Submit an SPR to DEC. The new 8 software support group says it is trying to look at all SPRs so give it a try. [RH]

HELP - 1970 DIBOL

Mr. Graham C. Killens writes:

"I was given your name by Mr Alastair Windrum of the UK 12 BIT DEC USERS GROUP, who informed me that either yourself or Mr. Lawrence Eisenburg could possibly obtain a 1970 or thereabouts copy of DEC DIBOL for me.

"I am a computer profesional workwise but also own a PDP-8/I as a hobby machine. The configuration is:

PDP-8/I +4K (8K total)
High speed reader/punch
2 TU56 (one unit)
2 TU55 (two units)
TC08N

The teletype boards were modified to allow a 30cp printer, VDU, and reader/punch.

My requirement is for a commercial type record handling system to assist local clubs in membership control, etc. As COS 300 only runs on 16K of memory and disk (?) I turned to 1970 DIBOL. I have some information in the form of DEC Manuals and DECUS DIBOL as well, but no software!

DEC apparently does not have 1970 DIBOL as it was superceced, niether do the user group. I am therefore hoping that some amateur or organization across the Atlantic does! From the little I have read DIBOL seems to be ideal and readily usable.

"I would be gratefull if you could research the possibilities of gaining DIBOL for me, and advise me of the outcome."

Mr. Killers address is 9 Meadow Way, Black Notley, Braintree, Essex, England. The original DIBOL was in the DECUS library for a time around 1970-71 but DEC withdrew it when they decided to rework it and reenter the commercial market with the COS 300 system. For this reason, it is no longer available from DECUS but I think it is perfectly legitimate for anyone who has a copy to share it with Mr. Killers. Incidently, a more modern and flexible solution might be to use OS/8 with one of its languages such as BASIC or one of the FORTRANs. Take a look at INVENT-8 in the DECUS library for some interesting routines and a sort that will work with FORTRAN II. Also, the excellent WVU sort/merge/extract package for OS/8 will be submitted to the DECUS library soon (I hope). [RH]

TERMINAL SWITCHING BUG

Kenneth M. Johnson sent the following note about a problem he is having when he switches his terminal between lines:

"I read, with great interest, the letter you received from Aaron S. Weg in the March, 1980 (#39) issue of DECUS 12 BIT-SIG Newsletter. His problem with the PDP-8 - LA120 interface seems similar to mine: We have a PDP-8/A connected to a LA36 DECwriter. Alternately, we have a switch on the LA36 that allows use of the DECwriter as an interactive terminal for the campus-wide DEC-10.

"When the LA36 is changed from either use to the other, (we experience a) lingering 'crash' of our FORTRAN Run Time System (FRTS Vers. 5A). This can only be rectified by re-booting the PDP-8. Obviously, for long runs, this is an extreme inconvenience.

Confounding this problem is the scarcity of support for the '8' in this area (indeed, my calls have been routed to Santa Clara, California, which can run up quite an expense), and the fact that hardware people perceive a hardware problem, while software people tend towards software interpretations. Our set up includes:

8/A CPU with 32K
RL01 Disk
LA36 with 20ma
VT50 terminal

"I am anxiously awaiting your reply."

Mr. Johnson is with the College of Liberal Arts, University of Arizona, Tucson, Arizona 85721.

"Instant analysis:"

I am not quite sure if the LA36 in this case is connected as an auxiliary device or if it is the standard console terminal (i.e. device codes 03 and 04). Presuming that the switching has been arranged correctly so the line does not "run open" when the terminal is not connected, the most likely explanation of this problem is predicated on the LA36 being connected to an interface that uses other than the standard console terminal device codes (i.e. 40 and 41 for example). Each time the terminal is switched from one circuit to the other, there could be a transient on the line to the computer that the interface would treat as an input character from the keyboard. This would set the input flag. That flag stays set until the computer is restarted or the appropriate instruction is executed. Ordinaraly, that is OK, but when FRTS starts, it

enables interrupts and the set flag starts causing interrupts. As supplied, FRTS does not know about special devices like auxiliary serial ports, so its interrupt service routine cannot find and clear the offending flag. The result is as described. FRTS will not work but non-interrupt driven programs (and OS/8) are OK.

If this is the problem, there are two solutions. One is to rework the terminal switching to eliminate the source of the transient that starts the problem. The other solution is to read some little known material in the OS/8 FORTRAN IV Software Support Manual (DEC-S8-LFSSA-A-D). On page 4-15 is information on two areas in FRTS where flag clearing instructions may be inserted in your copy of FRTS. One area is executed once when FRTS starts. This will clear flags that might have been set sometime in the past. This would fix problems due to switching the terminal before a program was started. (The other group of instructions is executed after each call to an OS/8 device handler (as contrasted with the internal interrupt driven handlers for TTY, LPT, etc.). Instructions in this area will clear any flags that are set as a result of actions of an OS/8 device handler that leaves a flag set when it exits, but instructions here will not help if the interrupts are being caused by the situation described above.)

If it is desired to switch the terminal while a FORTRAN program is running, the best solution is to look in the same chapter for information on the internal interrupt driven handlers (including a more or less correct listing). With that information and a little looking around in FRTS with ODT or FUTIL, it is possible to patch instructions right into the interrupt service code of FRTS that will look for and clear this sort of flag every time there is an interrupt. [RH]

FROM RON LARKIN

"I have written a new duplicating program for the Sykes 7250 flopppy disk unit. Features include (1) it copies entire IBM-format single density disks by bytes, not caring how the 12-bit words of the PDP8 are arranged; (2) after writing on the output disk, it compares against memory to check the validity of the written data; (3) when the program is saved as SYS:RXCOPY.SV, the CCL .DUPLICATE command works. It runs on an 8/E with EAE, but could easily be modified to use different hardware. Send a floppy to Ronald P. Larkin, Box 223, The Rockefeller University, New York, N.Y. 10021, and specify the packing format you prefer."

NOTE FROM DRS. F. J. MEIJER

"Referring to the questions about the programs DROP and RECOVR I inform you that the program DROP was written by J. Verburg at the time he was working at the T.H. Delft. The program RECOVR was written by F. Anthoni, rewritten by J. Verburg.

"The program DROP has been debugged by me, it now inserts the name of the ASCII-files if the user has made as first comment in these files:

```
/name.ex
```

"I assume names of non-ASCII-files are unimportant, as the user usually has a copy of these files. Those interested can obtain a copy of the programs from me."

The address is DIGICOS, Digitale Commerciele Systemen, Aart V.D. Neerweg 31, Ouderkerk A.D. Amstel, Postbus 24.

Firstly, thanks very much to the people who gave me so much help getting my KT8/I working, and to the person that sent me a copy of the Fortran IV optimiser which makes a vast improvement to my FIV code !!!

Why is it not possible to distribute all DECUS programs on the most commonly acceptable media, i.e. papertape ? I wish to order the following items but they are only available on DECTAPE ! Alternatively is there someone in the London area who could carry out an EPIC dump of some OS8 dectapes for me ?

8BAL (8-530)
FUTIL (8-608)
ADVENTURE (8-889)

Finally, if you use OS/8 Basic on a large m/c (say 32k) but find it difficult to write large programs because you keep getting the 'TS' error (too many characters in the string literal pool), take heart. I ran into this problem with a rather large program where I work, which generated 16k of pseudo code ! and I got around it by writing a new version of BLOAD.

BLOAD currently only allows 2800 (dec) locations for the system head, symbol tables, numeric variables, numeric literals and string literals, so if your program is at all verbose this determines the maximum program size rather than the core available.

The new version of BLOAD assigns space for the string literals, in a different area, but writes them to the end of the temporary file generated by FCOMP. BLOAD later reads the literals into the assigned core space, after loading the pseudo code. There is still a restriction on the total number of string literals, but not on their content, subject of course to the system length limit of 72.

I may in the near future submit this version to the library , but in the mean time I can send out copies of the binary for trial. It currently works under the V3 version of the OS/8 extension kit but should be OK with later versions.

My address is:-

R. Selby,
Flat 1,
3 Larpent Ave.,
Putney,
LONDON SW15.
ENGLAND.

SCOPE TYPE RUBOUTS IN FRTS

When the command "SET TTY SCOPE" is issued, bit 4 of 17726 is set, changing this system HLT to CLA HLT. This bit may be interrogated, as it already is by FUTIL, to implement scope-type rubouts in other programs. The patch below does this for FRTS. It uses the last 31 locations of the linemprinter ring-buffer.

FRTS V 5A

```
3072 / 4336 4774        JMS I SCRBAD    /WAS JMS ECHO
3073 / 0134 7000        NOP             /WAS BACKSLASH

3174 / XXXX 7550        SCRBAD, SCRUB

7546 /    0 6600        LPBUFR, 6600    /POINTER TO START OF
                                        /LP BUFFER: =6600 IF
                                        /USING MY PATCH FOR
                                        /HI-SPEED TERMINALS
                                        /(#33, P.17).
                                        /(ORIGINALLY 5166)
7547 /    0 3136        ECHAD,  ECHO
7550 /    0    0        SCRUB,  0
7551 /    0 6211        ST,     CDF 10
7552 /    0 1774                TAD I K7726
7553 /    0 6201                CDF 0
7554 /    0  375                AND K200
7555 /    0 7650                SNA CLA
7556 /    0 1376                TAD DIFF
7557 /    0 1377                TAD JSCOP
7560 /    0 3351                DCA ST
7561 /    0 5351                JMP ST
7562 /    0 4747        SCOP,   JMS I ECHAD
7563 /    0   10                10      /BACKSPACE
7564 /    0 4747                JMS I ECHAD
7565 /    0   40                40      /SPACE
7566 /    0 4747                JMS I ECHAD
7567 /    0   10                10
7570 /    0 5750                JMP I SCRUB
7571 /    0 4747        NOSCOP, JMS I ECHAD
7572 /    0  134                134     /BACKSLASH
7573 /    0 5750                JMP I SCRUB
7574 /    0 7726        K7726,  7726    /SCOPE BIT ADDR
7575 /    0  200        K200,   200
7576 /    0    7        DIFF,   NOSCOP-SCOP
7577 / 6600 5362        JSCOP,  JMP SCOP
```

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

Medical Center

150 Muir Road
Martinez, CA 94553

Veterans Administration

October 21, 1980

Mr. Robert Hassinger
Coordinator - 12 Bit SIG
c/o DECUS MR2-3/E55
One Iron Way
Marlboro, MA 01752

Dear Bob:

This note is a follow-up on a request for help we made in 12-bit SIG newsletter #34 (May 1979).

At that time, we had a problem getting our DMM 8-30 programmable memory management unit up. In response to that letter, the manufacturer, Digicos of Amstel, Netherlands contacted us and volunteered to fix the device if we returned it to them. We did, they did, and I am happy to report that we are so pleased with it that we intend to purchase another for our second PDP8E. We are running a modified RTS8 supporting 3 OS/8's (DECUS Proc. V.5, No.2 (Fall 1978)) but similar benefits would be realized by anyone running OS/8 under RTS8.

The numbers below are Fortran II compile-times for an arbitrarily selected sample program.

Stand-alone:	42 secs.
RTS8 DMM 8-30:	49 secs.
RTS8-KM8E:	186 secs.

These figures were generated with the second and third OS/8's idle. With both compute-bound, compile-times are as follows:

RTS8 DMM 8-30:	141 secs.
RTS8-KM8E:	552 secs.

This piece of hardware is a very cost-effective way to improve the performance of PDP8's. In our case, we are able to shut down at night (energy, you know) when the machine was previously on 24 hours a day.

Sincerely,

Hans von Blanckensee

In Reply Refer To:

DECUS

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

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.