



DECUS OS/8 SIG NEWSLETTER

July

Number 13

1975

Contributions and correspondence should be sent to:

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

REVIEW OF THE SPRING DECUS SYMPOSIUM - OS/8 Special Interest Group Activities

Most of the material in this item comes from a report written by Doug Wrege for the call for papers for the Fall U.S. Symposium.

An increased effort was made at the Spring Symposium to coordinate panel discussions and workshops following groups of papers in a given area. There was a time-sharing panel, a high level languages workshop, traditional PDP-8 family workshop, and the familiar OS/8 Special Interest Group meeting. There was a great deal of interest in time-sharing or multi-user or foreground/background type applications, both West Virginia University and Georgia Tech reported on work in this direction.

In the languages session a new intermediate level language called BLIP was discussed. It is somewhat like a simple version of BLISS-11. It gives an intermediate level language between something like FORTRAN and something more like PAL8. The generated code is PAL8 code. There was also a great deal of discussion and interest in OS/8 FORTRAN IV. Enough to possibly warrant a separate session for it at the next meeting. There was a Traditional Products session which consisted almost entirely of pre PDP8e users of PDP8's and PDP12's. A very active and interactive session with Bob Reed from DEC Traditional Products was the result. There was much interest expressed in forming a Special Interest Group to represent the traditional computers within DECUS. It has been concluded that for the moment this group can be a part of our OS/8 Special Interest Group because almost all current Traditional Products are in the 12 bit families. Therefore, I have included the first input from N. S. Kendrick who has been asked to act as coordinator for these activities to get the group going. Old timers will remember his activities going back many years, to the days of the original PDP-8. We plan to initially publish this information in our Newsletter, along with all other PDP-8 and PDP-12 related information. The trend seems to be away from strict concentration on OS/8 and more towards representing the entire spectrum of PDP-8 and PDP-12 problems and interests. Your inputs on this are solicited. Is this trend go, bad, or does it make no difference to the future strength and viability of the OS/8 SIG?

July 1975 Newsletter

Before leaving for Miami we were afraid there would be very little equipment from the PDP-8 and PDP-12 family. We knew only that we would have available a small PDP-8e with TD8e DECTapes for the sake of tape duplication. That machine arrived but much to our surprise Traditional Products arrived with a newly configured PDP-12 that they used to announce that the PDP-12 had become a Traditional Product and that they intended to support it. The users found the 12 very useful. After hurried calls back home we got LINCTapes sent to Miami (no one had thought to bring any) so that we had some software to run on the 12.

A COS-310 system also appeared. Some of the DEC system software people and a couple of users got together and worked up a demonstration of a very small network using some of this hardware. A pair of high speed serial interfaces appeared and were plugged into two of the PDP-8's to connect them. Some software was installed on both machines to implement a minimal "DECNET" between the two machines. RTS8 was run in both machines with OS/8 running in the background in each case. The machine with only DECTapes utilized the network to access one of the floppy disks on the COS-310 system and use it as its system device. After that the interruptable TD8e DECTape handler called SD8 (see last Newsletter) was modified into an RTS8 task and installed on the TD8e system and it demonstrated its ability to utilize the TD8e DECTapes while running under RTS8 and communicating on the network. Incidentally, this network actually ran faster than the PDP-11 network that DEC was officially demonstrating at the meeting. The reported reason for this was that there were some sort of bugs in the PDP-11 system, but the PDP-8 software worked first try.

HELP!

As you all know, this Newsletter is published too infrequently. The main reason is that it's a lot of work. If I had a regular input of ready to use items we could publish on a regular, frequent schedule. Individual items and regular columns can be submitted. Anytime there is enough material that does not require re-writing and editing to form the nucleus of an issue I will add what I can and get it to DECUS immediately.

There are a couple of guidelines to observe when writing articles for the Newsletter.

1. The DECUS Board has a non-commercialization policy that requires emphasis be on technical rather than commercial content. The Board also requires that prices not be given. However, it is all right to mention if a charge is made for an item.
2. Due to the high cost of printing and mailing we need to try to keep Newsletter articles short. DECUS underwrites all the Newsletter costs but as more SIG's form and publish Newsletters the DECUS Boards will have to take closer looks at the costs involved. If we all do our part to minimize costs we can keep this valuable channel of communication open and unrestricted.

July 1975 Newsletter

OS/8 SIG GROWTH

I recently reviewed the SIG mailing list and found that we have grown to over 1100 members. More than 200 of the members are in Europe. The availability of this information sorted by geographical area makes it much easier to help get Local Users Groups going.

The membership list is considered confidential information by DECUS. Access to it is restricted and requires a non-disclosure agreement but if you are interested in forming a Local Users Group contact me and we will work something out to get you the necessary information to get going.

FROM N. S. KENDRICK, JR. - TRADITIONAL 8-FAMILY S.I.G.

The interest shown at the Miami Symposium in updating the older family of 8 products led to the suggestion of a SIG devoted to efforts to maintain program capability across the product line. Considerable interest was also shown in acceptable hardware modifications to the 8/e group. For the present, at least, it would seem advisable to operate as a subset of the OS/8-RTS/8 group since these systems provide the main motivation for such work. I will try to serve as a clearing house for this information and pass on a digest to Hassinger for inclusion in these Newsletters. If those of you who tinker with the nuts and bolts of the operating systems will pass along your results to me maybe we won't all invent the same wheel too many times! (We may even come up with a round one once in a while.)

Work in our laboratories at present is directed toward the following goals, hopefully to be complete by the Fall meeting in Los Angeles:

1. Implementation of the BSW instruction on the plain 8 (can't call it classic anymore) and the 8/I-12 group. The 8/I seems to be very simple from a cursory examination of the prints.
2. A complete re-do of the memory extension-time share module for the 8/e incorporating a more intelligent trap as outlined at the Miami meeting. The design work on this is well under way.
3. Modification of the RTS/8 executive to make use of the new trap.
4. The trap modifications to the 8/I-12 group.

Your suggestions and help would be appreciated.

I have recently received a communication from J. Floris Anthoni of The Netherlands on their MULTI-8 real time foreground/time sharing background operating system which seems very interesting. As best as I can decipher Dutch his address is Medical Biological Laboratory TNO, Lange Kleineg 39, P. O. Box 45, Rijswijk, The Netherlands. This is a new hardware/software system similar to some of the ideas discussed at Miami.

July 1975 Newsletter

SWEDISH LOCAL OS/8 USERS GROUP ACTIVITY

Lars Palmer writes me that the Swedish OS/8 IUG is trying to construct a primitive library for primitive programs. What they intend is to construct a simple circulation list of programs and routines which are mostly not documented to the standard necessary to submit to DECUS which might be for other reasons not submittable to DECUS. These reasons might include such things as authors wanting to get a reimbursement for their work, or because there is some sort of bug in the program which makes it unsuitable for DECUS submission. They are presently experimenting with an unsophisticated circulation list sorted according to the DECUS category codes, and which includes information like programming language, cost bracket for the program, standard of documentation, and the author's name and address. It is up to the individual interested in the program to contact the author. They do not intend to do any kind of distribution themselves of anything except for the information on the list. This is a good example of how a Local User Group can provide help for the users in an area, and it also demonstrates how the Special Interest Group as a whole can help various users to exchange programs when they can't submit them to DECUS for whatever reason.

ADVANCED OS/8 SYSTEM DEVELOPMENTS

Since the last Newsletter the hopes for DEC sponsored OS/8 development beyond the currently planned maintenance update on OS/8, (which is supposed to include MACREL and its loader) seem to have dimmed due to lack of support among the various product lines at DEC. User interest seems to continue, however. It may eventually turn out that the users will end up doing their own development again if DEC doesn't get busy.

Just recently I received a note from Clyde Roby outlining some of his ideas for such a new system. Clyde's suggestions include the following:

1. It should be OS/8 compatible at least for ASCII sources.
2. The directory should be expandable (or at least expanded) beyond the present size.
3. The new system should be integrated into an RTS8 type environment.
4. It would require at least 12K,4K for the RTS8 part and 8K for the new monitor. It probably could require 16K of memory as most people can now afford that much memory with memory prices coming down the way they are.
5. An RTS8 environment allows many nice features such as (a) there should be a system task call to get/put characters to any device. It should be set up so that all devices would have the same call to get/put a character. This would make for very device independent I/O. (b) A spooling task may be inserted between the user call and the active device driver task to spool I/O. (Note that there has already been some work done on a line printer spooler which runs in the present OS/8-RTS8 configuration).

6. If the system were set up correctly an entire system should reside in a .SV file. Users could make specialized systems for their own different types of applications. They might be similar to the present /Y files but would be .SV files which would be "bootstrapped" too.
7. As far as command language syntax goes it would be nice to be compatible across DEC's main frame line as much as possible. (Note: DEC is presently working on defining a new command language standard which will be distinctly different from any of their existing command languages. More about this later.)
8. SCAN and WILD (vis-a-vis the PDP-10) could be incorporated into the RTS8 part of the system as an overlay task. Users could then implement some good programs by calling them via a system task call.
9. About security: As on other DEC software systems for the other main frames there should be programmer/project numbers, passwords, and file access words (read only, deletable, write only, etc.) similar to DOS-11, TOPS-10, etc.
10. The system should be as real-time as possible with an RTS8 type foreground but where TD80 DECTape and floppy disks are concerned true real time may be a luxury.
11. No time-share trap should be used! All system calls should be direct to RTS8.

Clyde is trying to get his version of LPTSPL ready soon but he says the work load at Georgia Tech is pretty heavy. Also, Clyde suggests there might be interest in a special category in the DECUS catalog for RTS8 tasks. Does anyone have any such tasks as yet and is there any interest in a special category of this sort?

MULTI-8

MULTI-8 is a real time foreground/time sharing background operating system for the PDP-8. Floris Anthoni recently sent me a copy of a manuscript which describes this system in over-view. He has been writing it over the last couple of years or so. It has much of the same real time, multi-task capability that RTS8 provides, and like RTS8 it also provides the ability to run OS/8 in the background. What is different is that the background OS/8 can be multi-user, that is there are multiple OS/8 jobs that are swapped in the background so that you effectively get a time-shared OS/8 system plus a real time foreground capability. One of the more interesting features of the foreground capability is that there are swappable tasks which are kept in a sort of a library on the disk and they are brought in when they are required. The most interesting feature here is that they are not required to load into specific core locations, as with the new version of RTS8 but rather the tasks are organized in a page relocatable way which is very efficient and very simple to use. The system allocates core dynamically at the time that tasks are loaded, and it frees the space used when the task terminates itself. I am hoping to be able to get more information on this system and to explore what sort of plans the author has for making it available before the next Newsletter.

STANDARDIZING FOREGROUND IOT's FOR OS/8 BACKGROUND PROGRAMS RUNNING UNDER RTS8 AND OTHER FOREGROUND/BACKGROUND OPERATING SYSTEMS

As more and more foreground/background type software is written and becomes available, there is an increasing need to agree on some sort of conventions for assigning the IOT codes that are used to make calls to the foreground or operating system. The same type of calls are presently made by the TSS8 (EDU-system 50) software. These calls are similar to the type of calls one might make to a larger computer's operating system, but they are implemented as psuedo IOT's in the typical PDP-8 system because the instruction trap is normally used to implement such a system and it provides the most convenient way to communicate from the background program to the foreground. Presently we have RTS8, MULTI-8 has been announced, OMSI is working in this area, EDUcomp is developing a similar sort of system, and various people are writing routines to go with RTS8 and other similar environments. In each case arbitrary selections are being made by the authors for the IOT codes. Obviously in the future there will be many conflicts unless we can agree on conventions. If you are working in this area, or have ideas, please send them along and we will try to coordinate and gain some sort of agreement on how to approach this issue.

CUSTOMIZING CCL FOR YOUR INSTALLATION

Lars Palmer has included in his most recent letter to me some information about how his installation has customized their version of CCL to suit their uses and their mode of operation. He thinks that this sort of information would be useful from many different users as each installation has their own special needs. His changes are as follows:

1. In their system (which is a disk system with a floating point processor) their main software is FORTRAN IV. To optimize FORTRAN IV usage they have made several modifications to CCL.
 - A. The TECO and MAKE commands default to FORTRAN (.FT extension).
 - B. A call to FRIS on the .LD extension has been added to the system.
 - C. The default mode alt-mode in the EXECUTE command has been removed. They find that it causes a lot of trouble because it leads to (1) the annoying forgetting of file specifications with FORTRAN IV and (2) he has had difficulties as mentioned elsewhere finding a way to pass an alt-mode in the BATCH file.
 - D. They have changed the order of searching for file name in such a way that the EXECUTE command always defaults first to the .LD then to the .RL and finally to the .FT extensions and then to the other file name extensions that are normally checked. This means that he gets the sort of operation that you would probably like best in a FORTRAN environment. If a .LD file exists then it is executed directly and so on down the line. If only a .FT file exists it will be compiled and loaded and executed, etc.

July 1975 Newsletter

- E. They have modified the COMPILE and EXECUTE commands in such a way that the default output file is on DSK:. This means that it is possible to do a COMPILE TEST followed by an EXECUTE TEST which will pick up the compiled version. What is happening here is that normally the default output file for some of these commands is SYS: rather than DSK: and in a system where these two names refer to different devices, you normally can't do what you would like. I find this true in my DECTape system in particular where the distinction between the two devices is the normal way of operating the system.
2. They have found the DELETE command is very dangerous, particularly in a disk system where it executes very quickly. Files disappear before you have a chance to think about what is happening. This is even more so where they have a 2400 baud console on which the file names go by very fast as they are output from the delete function of FOTP. Therefore, they have included the /Q option in the DELETE command so that it always forces you to answer "yes" or "no" to each delete. In order to allow for deleting in those cases where you are sure of what you want to do, such as in a BATCH file, they include a new command which they call KILLX which works the way the old version of the DELETE command works. Doug Wrege's newest version of DECsystem-8 has made a slight change to the DELETE command also. When you specify a delete of a single file (i.e., no wild card specifications) he goes ahead and deletes the file in the normal way. But when you specify wild card specifications so that considerable numbers of files might be deleted, then the /Q option is automatically forced.
 3. The DATE command has been implemented in such a way that it runs a special BASIC program which writes a TECO MACRO which includes today's date. Then when you do a TODAY file-name, TECO will do the normal EB on the file but with today's date inserted as a comment at the top of file. The default comment character is a C as in FORTRAN code, but any comment letter can be passed to the file. He includes a listing of the BASIC program with his letter. If anyone is interested in it let me know and I will send a copy. To make this work he has to inhibit the chaining test in BASIC. This is done by changing location 7001 in the BASIC save module to 7000 (NOP) with this change the program will work in all cases except if DATE is given under BATCH.
 4. As mentioned under the paragraph on EXPIP Lars has implemented UPDATE and MOVE commands which are very convenient in certain situations.
 5. Lars uses an off-line paper tape punch where paper tape is punched which is read in for revision later on the computer. They have a special handler which does some editing on the tape as it is read in. This is actually a modified version of the new 2-page teletype handler. To simplify reading the data in from the paper tape reader they have implemented a new command FETCH which will simply fetch a file from the

paper tape reader giving it the name which is the argument that goes with the command. It will use the special handler by default. If a /A or /B option is given then PIP will use the normal paper tape reader handler.

6. He has taken the PRINT command and arranged it so that it utilizes Clyde Roby's PRINTR program for formatted printing of files on the line printer.
7. Lars has also implemented a DUMP command which will output in octal form. Doing DUMP A,B will output a listing of the differences between files A and B. This is implemented using the program OCOMP which is available from DECUS as 8-608.

EXPIP

Lars Palmer from Sweden has submitted a new version of his EXPIP program to the DECUS Library. It is a PIP type program that has features that neither PIP nor FOTP have. It allows wild card specifications for file name extension in particular. It contains several features that are nice for people with two DECTape systems that make moving files around more efficient. It has a command that permits merging files which is sometimes very nice using wild card file name extensions. There's an option to transfer files only if there is not a file of the same name on the output device with a more recent or the same date as the input file and there's an option to never transfer a file if there is a file of the same name on the output device. There is an option to remove the input files from the input device after the transfer has been completed successfully. Another variation on this is to transfer files only when there is a more recent version than the version that exists on the output device. The Q option as in FOTP gives you a chance to decide on each individual file and you can specify today's date for the new output file. The quals option lets multiple copies of a file to be made. This means that a listing function could be implemented to produce several copies of a file on the printer automatically from one call. Lars says that using this program he has implemented two new commands under OS/8 version 3 and CCL - these are UPDATE and MOVE. These commands do what their name suggests, he says. UPDATE will write new copies of files on the output device that already exist there in older versions and MOVE will do what COPY does but after doing the copy it deletes the input file. I find with a limited DECTape system that these commands can be very useful.

I think we have finally solved the problem the DECUS Library was having with EXPIP also. There have been difficulties with tapes from Lars' part of the world coming through unreadable. Something has been happening somewhere along the line to some of them but the current tape seems to be all right.

SOFTWARE UNDER DEVELOPMENT BY LARS PALMER

Lars Palmer has written about several routines that he is working on. He hopes that some of them will be released to DECUS in the future when he has a chance

July 1975 Newsletter

to document them properly. In the meantime, he has sent along a short summary of what they are about and if anyone is interested they can contact him.

1. A small routine BGET and BPUT which allow you to utilize the 36 individual bits in a floating point word under OS/8 FORTRAN IV. The routines are called exactly the way CGET and CPUT are. This routine allows you to use a very large number of logical variables as in a program such as Conway's game of life.
2. A small routine called FILE SIZE which will communicate to the FORTRAN IV program the size of the different files given at run time. It will also tell you if a certain file number is associated with one of the internal handlers built into the run time system.
3. A number of small routines modeled on the PDP-12 routines available in the FORTRAN IV Library but written for the PIP-8e digital interface. Utilizing the routines in a special patch to the run time system he has also implemented the possibility of reading command decoder switches into the FORTRAN program. This makes it possible to modify the action of a FORTRAN program in a BATCH file. At the moment the communication is via the MQ register but Lars says that it's a trivial modification to change this.

Another program that Lars has functioning which he will make available to DECUS as soon as the documentation is produced is a program for filling in forms on a high speed VT05, a blank version of the form is written and kept in a file. The program displays the blank form on the VT05 face which then is filled in appearing write protected in all places except the fields which are to be filled. The filled in information is then written to a file and can be used for further editing. He says this gives a very fool proof way of inputting data to be used by other programs. The program is written in FORTRAN II and requires 12K and a high speed VT05. A disk is an advantage but not necessary. However, it must have a second logical OS/8 device besides SYS:.

SSP THE SCIENTIFIC SUB-ROUTINE PACKAGE

In the last few months there has been a lot of interest in the Scientific Sub-routine Package which is available on many computers. I understand it has been placed in the public domain by IBM. Apparently it exists in at least two different forms. One for the IBM 1130 and one for the IBM 360/370 family. Apparently the 1130 version is a subset of the full version. Dr. Arthur Stiemmon has written to tell me that he managed to convert the 1130 package for OS/8 FORTRAN IV successfully. He is planning to make his work available for a nominal fee, I believe. Tom MacIntyre also indicated that some work had been done in this same area at West Virginia although he has not mentioned any plans to make it generally available. Since then I have been talking to Dave Todd, who is heading up the efforts of the PDP-10 Main Frame Group to improve the PDP-10 part of the DECUS Library. As it so happened, Dave also is the most recent submitter of the SSP for the PDP-10. This is listed in the DECUS catalog

July 1975 Newsletter

as 10-101a now. The difficulty with that package for the rest of us has always been that it was only available on 1/2 inch magnetic tape in PDP-10 FAILSAFE format so it was not readily accessible. Dave has converted the package to 5 PDP-10 format DECTapes for me. I have converted the files to OS/8 format DECTapes (six of them!) and I have fixed some of the more obvious compatibility problems. If I can find enough DECTapes I hope to submit the package to DECUS later this summer.

NEWS FROM OMSI

Rusty Whitney from OMSI recently wrote me asking to mention in the Newsletter that a routine for reading BCD data from a half-inch MAGtape and also a routine to handle the XY8e plotter which works under OMSI's PS/8 FOCAL are both available from Harold Cronin, Code 602, Naval Weapon Center, China Lake, California 93555.

At the Spring Symposium OMSI also revealed that it has been working on a multi-user OS/8 system. No details are available as yet and I understand there is still development work to be done.

MATERIAL FROM JIM CRAPUCHETTES

Jim has sent along a couple of patches for the FORTRAN IV run time system. One of them allows you to control the INPUT ERROR condition which normally, when detected, causes an abort of the program. When you are writing a program that is designed to collect data from manual input and record it in a file this is a bad feature of the system because any input format error can cause the program to abort and then you end up losing all of the data that you input.

Jim also included a set of ODF patches which cause the FORTRAN run time system to ignore both the default input format specification and spaces in the input. He says that this allows truly free format input from the keyboard although it does not adhere strictly to the FORTRAN specification. Together the two patches make a nice data input package for keyboard input to FORTRAN.

Jim is in the process of making some changes to his version of TECO. He is going to try to make it compatible with the DEC version. His work is aimed at new features related to I/O that will make it more powerful than DEC's TECO. He plans on such things as multiple input files, square brackets specified length for the output file commands, and the ability to reset the input file to the beginning. He will also allow a ":" to modify a file command so it returns a succeed or fail code rather than giving an error. Jim is interested in suggestions for changes or additions in connection with this work. Anyone interested can reach him at Frelan Associates, P.O. Box 298, Menlow Park, California 24025. Jim has also included a fix for his program FUTIL which is in the DECUS Library.

July 1975 Newsletter

INFORMATION FROM REVEREND CHASE

Reverend Geoffrey Chase from the Portsmouth Abbey School wrote recently to make a couple of suggestions and to pass on a coding technique which he has found useful. First, he noted the error in the recent Digital Software News in a patch to CREF which was stated as a 7014 when it should be a 7012. Second, Reverend Chase suggests that in the future DEC should do something to make the length of the string that you can pass to DECO using the MUNG command longer. Apparently when the MUNG command was designed no one anticipated the need to pass longer strings, but it seems possible to do it.

Reverend Chase wondered about what the system does when you load a handler over a previously loaded handler. The documentation seems lacking on this point. As an example, if you were to load the LPT: handler over the DECtape handler what would happen. It turns out that the system is clever enough to know to delete all 8 entry points of the TCO8 handler from its residency table. Oddly enough very few people really know that this is true. Apparently it was put in the code back in the PS/8 days and it's been there ever since waiting till it was needed. Reverend Chase has passed along a neat coding trick which many people are not aware of for OS/8 FORTRAN II. As you know, OS/8 FORTRAN II integers are limited to 12 bits. Reverend Chase's coding technique gives a quick, easy, fast way to take the integer part of floating point numbers which of course can give a 6 digit magnitude. His demonstration program for computing prime numbers also gives a quick way to get a vertical tab on the teletype.

NOTE FROM BILL KAUFMAN

Bill Kaufman from Mobil Research and Development has sent along information which includes the following: A version of LIB8.RL which includes his software to use the PDP-8e EAE for subscripting, integer multiply and divide, 27-bit floating multiply and 23-bit floating divide (both using the standard FORTRAN II format). I/O device 1 obtains input from the batch file when batch is active. Device 3 supports I/O to an ADDS 880 page buffered CRT, and devices 5 and 6 represent TC58 tape drives 0 and 1. This handler writes 80,80 EBCDIC with blocking of only records that are less than 80 characters.

Bill has also sent a slow running cross-reference program written as a TECO MACRO for use on FORTRAN II programs which produces an interesting side-by-side type of listing with both the FORTRAN and part of the assembly code listed.

LETTER FROM ALBRECHT LOMMEL

Mr. Lommel from Zurich has written to tell us about some contributions he has made to the library and to inquire about the possibility of the availability of certain FORTRAN IV software. He has submitted four programs to the library. First, "FASTAD" fast data collection, "WDATA" for writing 12-bit binary data in the form of normal 256 word OS/8 blocks on the system device, "RDATA" to read this

July 1975 Newsletter

data under FORTRAN II and "TTYIO" the necessary I/O package for serving the teletype. He says he would like to know whether any of the FORTRAN IV users has software for the ADO1-AP A/D converter and whether they have changed the necessary modules in the FORTRAN IV package so that the real time capabilities for data acquisition can be used with this type of converter.

SUGGESTIONS AND COMMENTS FROM DR. G. M. HODGSON

Dr. Hodgson recently wrote with several suggestions and ideas. Among others he was interested in the idea of reviewing DECUS Library materials. He suggests a review sheet for use in reviewing program that would include the following information:

1. The reviewer's equipment configuration and version of the software used to assemble the DECUS program.
2. The experience of the reviewer.
3. Problems found with the program when compiling, loading, starting up, running, input/output, and accuracy results.
4. Restrictions found that are not noted.
5. Comments on adequacy of write-up.
6. Corrections, suggestions, patches, additions, changes to the program or write-up.
7. Recommendation for retention in library.

He suggests that if this information is to be valuable it needs to come from a broad base. Once a sufficient number of replies for a particular program have been received, a summary evaluation form could be prepared. It should indicate how many persons had evaluated the program. This would be attached to the program write-up. Once the review process was underway an evaluation sheet would be sent out with each program when ordered. Periodic updating of the evaluation would be included. Alternatively the evaluator could request his comments be added to the summary sheet if he thought his problem with the program was severe. If the evaluation indicated the program was to be eliminated from the library, the author would have the option to correct and re-submit it. An evaluation process might limit the number of programs submitted but Dr. Hodgson thinks that they would be of higher quality. Some of this suggestion parallels earlier efforts but perhaps with new emphasis it could be made to work better than it did in the past. Some of his ideas may or may not be popular. I think though that this is a good place to start on gathering ideas on the subject and I'm still soliciting more inputs from everyone who has an interest.

July 1975 Newsletter

Dr. Hodgson also suggests that we need a process to alert users to updates of programs. The cost of such a review process might increase the price of the DECUS service charges, he thinks. However, he suggests that if the result is a more reliable program that the higher cost would be offset by less time spent getting the program to operate. This idea comes up from time to time, that of asking DECUS to find a way to alert people who have previously received a copy of a program to updates of it. If you find that is something that you think would be valuable for the library to try to do, let me know. It would require considerable effort on the part of the staff that operates the library and would most likely increase costs of operation of the library substantially, but it's possible that there are enough users who want this service so that it should be considered.

Dr. Hodgson has made several suggestions regarding improving the quality of the write-ups that go with library programs. First, he would encourage including a flow diagram with the listing so as to enable users to understand how the program was designed and aid in locating the problems as they occur. And second, if the program makes calculations the author should include a test problem with its results. This will speed acceptance in checking the program by the new user. I find flow diagrams are hard for some people to prepare and some programs would be quite difficult to fully document in that way, although certainly we should encourage it wherever possible. Do you think that requiring flow diagrams is worthwhile considering that such a requirement might discourage submissions? I certainly agree with Dr. Hodgson on the second point. Authors should provide a generous supply of examples and tests that a user can run to verify that the program is working and to see how it is intended to be used, at least for some simple test cases. Many of the widely recognized OS/8 submissions in the library have such examples and I think the popularity of the programs have been enhanced by the availability of the examples.

The final area that Dr. Hodgson is interested in regards the problems of a new user. He finds, for instance, that as a new user it's hard for him to discover how PAL8 difference differs from PAL3, PALD, or PAL10, or how FOCAL 8 differs from FOCAL 1968, FOCAL 69, or FOCAL W, or even how OS/8 differs from PS/8. Also learning things such as the difference between a PDP-8 and the earlier PDP-8's. He needs to know what sort of software differences there are between the machines. He says that he's not aware of any source of this sort of information although he suspects that some of it might be available in a DEC training course, but in many cases people don't get to go to those training courses because of the way they purchase their systems, or because they aren't the first person to use the system, and the training credits have been used up. Basically the problem here is help for the new user. This seems to be an area that the Special Interest Groups can help with considerably. What we could do is to ask for volunteers in various parts of the country to offer a little bit of help, perhaps on the phone, to new users who need it and to make information available to new users as to who the volunteers are who will give them advice and answer their questions. It seems that this is useful judging from the experience of other user groups. Supposedly DEC software support gives this kind of help but, in general, it doesn't work out too well because the software support people really don't seem to know that much about OS/8 either. As we hopefully build up some Local User Groups within the OS/8 Special Interest Group they could be

July 1975 Newsletter

focuses for this sort of help. If we get some volunteers we may be able to name a HELP coordinator and that part of the OS/8 SIG might choose to sponsor or organized a tutorial session at each DECUS Symposium to help the new users. This is now being done in the case of the PDP-10 sessions, for instance.

NOTE FROM I. M. TEMPLETON

Mr. Templeton has sent a short note which I will attach to the Newsletter which he thinks may be of interest to OS/8 users who have a simple cassette or tape unit for backup storage. The device he refers to in his article is a Progressive Systems "ZIPPER" cassette unit for which he has record and replay device handlers available also.

SPR's FROM LARS PALMER

Lars Palmer has written with a summary of some SPR's that he had submitted to DEC from time to time. He has not received answers to these SPR's.

1. It is impossible to pass an alt-mode (dollar sign) from the EXECUTE command under BATCH to a FORTRAN IV program. This is a restriction of the system which DEC has answered but Lars is not satisfied with their answer. He says that it means that it is impossible to run a FORTRAN program under BATCH which requires file input. (Note: Check Digital Software News, April '75, for possible solution.)
2. Certain restrictions exist in CREF when it is used with RALF input. In RALF it is possible to have a symbol followed by a comma in places where it is not the definition of the symbol. CREF misinterprets this and will take the latest such reference to be the defining of the variable.
3. There are some problematic restrictions in RALF. Certain codes will not assemble properly. In some cases RALF will assemble a two-word reference when it should assemble a one-word reference. It will also sometimes give a redefinition error when there is no redefinition involved. He has sent an SPR on this question but has not received an answer. (Note: Some of this problem may be explained by an obscure note in the OS/8 Handbook at the bottom of page 5-22. There they tell us that the location of the base page via the BASE pseudo op and the location of operands using ORG= must be defined in the coding before an FPP instruction refers to them. When this is true then the short form of the instruction is assembled, rather than the long 2-word form. Essentially this seems to mean that all of your variable storage wants to be defined and your base register area wants to be defined before the code that references it. This "feature" of RALF can be most puzzling to a programmer who is accustomed to an assembler like PAL8 where it doesn't matter whether the symbol is defined before

July 1975 Newsletter

or after the instruction that refers to it. It's particularly surprising because the information that gives you the clue to what is going on is so well hidden in the documentation. Of course this is true of many aspects of RALF and I hope we can encourage DEC to do something about that.)

4. If, by mistake, two inputs files are defined to the FORTRAN IV compiler (when the first one is the only one required) the compiler halts. The solution is a simple patch which follows:

```
6474/XXXX 7700
6433/1255 1274
```

5. Lars has problems with the FORTRAN II loader which refuses to load properly if there is a software core size set in the machine. As OS/8 BATCH sets the software core size equal to the BATCH field it means that he cannot do a FORTRAN II load under BATCH at the present and he does not understand why. (Check Digital Software News for possible answer.)

SPR REGARDING FORTRAN II LINKING LOADER

This SPR was submitted by Albrecht Lommel the 20th of March 1975. The problem is that a FORTRAN program which runs under OS/8 version 2 correctly does not run with the new release - that would be OS/8 version 3. The source of the program has been compiled and reassembled under the new software. The difficulty is that the new release of the loader does not seem to load the program correctly. The new program requires more than 8K and the difficulty seems to be that the loader is not correctly recognizing more than 8K of core. He says that this version of the FORTRAN II loader was not able to form a correct core control block. It was necessary to get a map to insert the explicit arguments for the monitor SAVE command to get a correct saving of the program on the system device. This seems to continue to be true in spite of the fact that the patches for the FORTRAN II compiler SABR and LOADER published in the February 1975 Software News have been made. (Note: Digital Software News, July '75 FORTRAN II #3 might fix this.)

SPR REGARDING DIFFICULTY WITH OS/8 BATCH ON A 32K SYSTEM

This SPR was submitted on the 29th of April, 1975. When OS/8 BATCH is run on a 32K system it sets the software core size to the top of field 6 rather than the top of field 7. The first problem is that BATCH cannot utilize the entire 32K system but will only utilize 28K of such a system. The second problem is that when BATCH terminates it leaves the software core size set to the top of field 6 rather than restoring the full 32K capability of the machine. Thus, the only way to recover the use of the rest of the core is to use a CORE command to regain the last space. This must be done manually. The CORE command is illegal under BATCH.

July 1975 Newsletter

I have just received an answer to this SPR dated 26 June 1975 (9 weeks!). It gives a temporary fix. If anyone needs it let me know.

SPR REGARDING A BUG IN THE OS/8 FORTRAN IV LOADER

I submitted this SPR to DEC on the 29th of April and received an answer very quickly. Apparently the article that they sent back as an answer was already in preparation at the time the SPR was received. The answer is scheduled to be published in the July, 1975 Software News. The OS/8 FORTRAN IV loader version 22 does not obey the software core size correctly.

SPR REGARDING DOCUMENTATION OF OS/8 FORTRAN IV STATEMENT FUNCTIONS

This SPR was submitted on the 29th of April, 1975 and no answer has been received to date. The OS/8 Handbook states that arguments to statement functions are "dummy" arguments and then goes on to say that they should not be referenced outside the statement functions. Actually what happens is that the arguments are not what we would commonly think of as dummy arguments. The name used refer to the same variable as the name refers to in the rest of the program. In other words, the "scope" of the arguments includes the entire program rather than just the statement function. This appears to be inconsistent with the FORTRAN IV standard. You should be careful when using statement functions to make sure the "dummy argument" variables are referenced no where except in that statement function, or else you should check the generated code to be sure that nothing is happening that you had not anticipated with respect to conflicts in the naming of your variables and their storage.

SPR REGARDING THE FORTRAN IV COMPILER MAKING ERRORS WHILE COMPILING STATEMENT FUNCTIONS

This SPR was submitted 29, April 1975 and no answer has been received to date. The FORTRAN IV compiler seems to make compilation errors under certain conditions when it is compiling statement functions. In an example that was included with the SPR there was a statement function definition which included a call to a function from the library. One of the arguments given to the library function was an argument that had been passed to the sub-routine. If you study the compiled code very carefully you discover that the internally generated location of tags gets confused and the program ends up branching to an incorrect location. It appears that this problem can be avoided by setting a local variable equal to the argument passed to the sub-routine giving you a local copy of the variable. In this case it appears that the compiler does not make the same error.

SPR REGARDING THE FORTRAN IV COMPILER

I submitted this SPR to DEC on the 29th of April, 1975 and no answer has been

July 1975 Newsletter

received as yet. This SPR involves a very subtle problem that took many hours to track down. Under certain conditions the OS/8 FORTRAN IV compiler generates code involving EXTERNAL type variables incorrectly. When the EXTERNAL type variable is passed as an argument to a sub-routine, code which uses the value of that EXTERNAL variable may be incorrect and cause the system to blow up when the EXTERNAL routine is called. The conditions under which this happens are when the compiler places the storage for the external type variable after the sixth variable in the base register area. Those who have looked carefully at the assembly code generated by the FORTRAN IV compiler, and who understand how the FPP code works know that in sub-routines six variables are placed in six of the first eight base registers and then the balance of the variables are pointed to by the registers above that, the difference being that the FPP can use the first eight registers in ways that it cannot use the rest of the registers. For this reason, the code that references the first eight registers works differently than code that references the other registers. The compiler allocates those first six registers first for arrayed type variables because those are the only registers that can be used to access arrayed type data. And, then after all of the arrayed type variables are taken care of then the compiler starts taking simple, undimension variables, in order that they are declared in the argument list. In order to avoid the problem that this SPR reports it seems necessary to have the total of the dimensioned variables named in the argument list, plus the number of EXTERNAL type variables total no more than six, and the EXTERNAL type variables should come before any other nondimensioned variables in the argument list. If you think you have a problem with this and you don't see an answer to it from DEC but you don't really understand contact me and I will try and make it clearer.

OPERATION OF BLOCK-ORIENTED, NON-FILE-STRUCTURED
STORAGE DEVICES UNDER OS/8

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

It appears that the most satisfactory way of sensing an end-of-file condition on playback from a block-oriented, non-file-structured device is for the data to terminate before the playback buffer is filled. The rest of the buffer is then filled with zeros and a non-fatal error return taken. This approach necessarily rules out use of the /A (Ascii) or /B (Binary) modes under P1P, except for very short files, since the playback buffer is larger in both cases than the record buffer. This results in automatic termination after one output buffer is replayed.

The /I (Image) mode is not subject to this limitation since the size of both record and playback buffers is the same (6400₈ words, or 13₁₀ OS/8 blocks). In this case, however, another problem occurs if the stored file is any multiple of 13₁₀ blocks in length (e.g. CREF.SV), since no unfilled buffer end-of-file condition can occur. The system then continues to read subsequent files until a terminating condition is found. This problem can be overcome by artificially increasing the size of any such file by the use of the /I=n option under P1P. The command *CREFA.SV<CREF.SV/I=16 creates CREFA.SV, 14₁₀ blocks in length, which can be recorded and replayed correctly with no change in performance.

/"INPUT ERROR" CHANGE FOR FRTS

/BY:

/ JIM CRAPUCHETTES
/ FRELAN ASSOCIATES
/ P.O. BOX 298
/ MENLO PARK, CALIF. 94025

/ THIS PATCH FOR VERSION 3 OF FRTS (THE RUN-TIME
/ SYSTEM FOR OS/8 FORTRAN IV) CHANGES THE ACTION OF
/ AN INPUT ERROR (ILLEGAL CHARACTERS IN THE INPUT
/ STRING) WHEN THE INPUT DEVICE IS THE TELETYPE AND
/ THE LIBRARY FUNCTION "CHKEOF" HAS BEEN CALLED.
/ UNDER ALL OTHER CONDITIONS, THE "INPUT ERROR" MESS-
/ AGE IS OUTPUT, FOLLOWED BY AN ERROR TRACEBACK AND
/ ABORT TO OS/8.

/ USAGE OF "CHKEOF" WITHOUT THESE MODIFICATIONS
/ IS DESCRIBED ON PAGES 8-50 AND 8-51 OF THE OS/8 HAND-
/ BOOK. AS WITH THE STANDARD VERSION, "CHKEOF" MUST
/ BE CALLED BEFORE EVERY FORMATTED READ TO ALLOW TEST-
/ ING FOR THAT READ. AFTER THE READ, THE CONTROL VAR-
/ IABLE WILL INDICATE WHAT HAPPENED DURING THAT READ
/ AS FOLLOWS:

VALUE	MEANING
-	ERROR IN INPUT STRING (TTY INPUT ONLY)
0	ALL INPUT DONE CORRECTLY
+	EOF (CTRL-Z) FROM INPUT DEVICE

/ WHEN THE VALUE OF THE CONTROL VARIABLE IS NON-
/ ZERO, THE VALUES OF ALL VARIABLES READ SHOULD BE
/ CONSIDERED TO BE QUESTIONABLE.

/ SEE THE NEXT PAGE FOR A SHORT EXAMPLE PROGRAM.

/"INPUT ERROR" CHANGE FOR FRTS

PAL8-V9B 06/05/75 PAGE 2

```

/C PROGRAM TO TEST CHANGE TO INPUT ERROR HANDLER
/
/100 WRITE (0,1100)
/1100 FORMAT ('0INPUT VALUES'//)
/
/ SUM= 0
/ DO 290 I=1,1000
/210 WRITE (0,1110) I
/1110 FORMAT (I5,2X$)
/ CALL CHKEOF(IERR)
/ READ (0,1120) VALUE
/1120 FORMAT (G14.6)
/
/C IF IERR = 0, ALL OK. IF IERR < 0, INPUT ERROR, SO RETRY.

/C IF IERR > 0, EOF (CTRL-Z) SO INPUT DONE.
/
/ IF (IERR) 210,290,300
/290 SUM= SUM+VALUE
/
/300 AVERAG= SUM/(I-1)
/ WRITE (0,1210) SUM, AVERAG
/1210 FORMAT ('0SUM = ',G14.6,' AVERAGE = ',G14.6//)
/ GO TO 100
/ END

```

/"INPUT ERROR" CHANGE FOR FRTS

PAL8-V9B 06/05/75 PAGE 3

/OVERLAYING FRTS WITH THESE PATCHES

/

/ 1. ASSEMBLE THIS SOURCE FILE WITH PAL8

/

/ .R PAL8

/ *INERR<INERR

/

/ 2. OVERLAY USING ABSLDR

/

/ .R ABSLDR

/ *DEV:FRTS.SV/I

/ *INERRS (THESE PATCHES)

/

/ 4. UPDATE FRTS VERSION

/

/ .ODT

/

/ 5534/ 4040 3040 (CHANGE " " TO "X ")

/

/ *C

/

/ 5. SAVE NEW VERSION

/

/ .SA DEV:FRTS 0-7577,12000-13777,15400-17577;200=2000

/DEFINITIONS FROM FRTS:

0016 VEOFSW= 16

0025 EOLSW= 25

0034 ERR= 34

0036 MCDF= 36

0100 HAND= 100

0140 LIT40= 140 /ACTUALLY A PAGE 0 LITERAL

0320 TTY= 320

/'INPUT ERROR' CHANGE FOR FRTS

PAL8-V9B 06/05/75 PAGE 4

/ACTUAL CHANGES TO FRTS:

/CHANGES TO I,G,E,F INPUT CONVERSION ROUTINES

2435 *2435 /'INER' - OLD ERROR CALL, ILLEGAL CHARACTER
~~02435~~ 5240 JMP INER1 /GO A SAFE PLACE TO CALL NEW ERROR

2440 *2440 /'INDCPT'+2 - TWO DECIMAL POINTS SEEN
~~02440~~ 4770 INER1, JMS I AINER1 /CALL NEW ERROR - SAFER PLACE

2506 *2506 /'INE'+1 - TWO "E"S SEEN
~~02506~~ 5240 JMP INER1 /GO CALL NEW ERROR

2570 *2570 /'INESW'+1 - WAS FREE
~~02570~~ 7400 AINER1, INERX

/CHANGE TO L INPUT CONVERSION

2630 *2630 /'LINGCH'+11 - ILLEGAL CHARACTER SEEN
~~02630~~ 4763 JMS I AINER2 /CALL NEW ERROR ROUTINE

2763 *2763 /'HCW'+1 - WAS FREE
~~02763~~ 7400 AINER2, INERX

/"INPUT ERROR" CHANGE FOR FRTS

PAL8-V9B 06/05/75 PAGE 5

/CHANGE ADDRESS OF INPUT ERROR CALL IN ERROR LIST

4614 *4614 /WAS 'INER' - ADDRESS CORRESPONDING TO
 04614 7407 INERR / "INPUT ERROR"

/CHANGE LINK WORD TO 'LPBUFE'

7371 *7371 /AT END OF 'LPBUF4' (PLAIN VERSION)
 07371 7422 LPBUFE

/ADD NEW ROUTINE AT THE BEGINNING OF THE LPT RING
 / BUFFER SECTION. REQUIRES THAT THE LINKS TO THIS
 / SECTION ALSO BE CHANGED TO NEW BUFFER START.

7400 *7400

07400 0000 INERX, 0 /SPECIAL HANDLING FOR INPUT ERROR
 07401 7200 CLA
 07402 1100 TAD HAND /IS THE INPUT THRU THE TTY?
 07403 1221 TAD MTTY
 07404 7650 SNA CLA
 07405 1016 TAD VEOF SW /YES, HAS SWITCH BEEN SET?
 07406 7450 SNA
 07407 4434 INERR, JMS I ERR /NOT TTY OR NOT SET UP
 07410 4436 JMS I MCDF /TTY & SET UP, MAKE A CDF
 07411 3212 DCA .+1
 07412 7402 HLT /EXECUTE THE CDF TO THE
 07413 7240 STA / FIELD OF VARIABLE AND
 07414 3417 DCA I VEOF SW+1 / SET MANTISSA TO 7777,0
 07415 6201 CDF 0 / WHICH IS NEGATIVE.
 07416 1140 TAD LIT40 /NOW SET 'EOLSW' TO A SPACE,
 07417 3025 DCA EOLSW / FORCING REST OF LINE TO
 07420 5600 JMP I INERX / SPACES ('EOLINE' RESETS IT)
 07421 7460 MTTY, -TTY

7422 LPBUFE= . /NEW START OF BUFFER SECTION

/"INPUT ERROR" CHANGE FOR FRTS

PAL8-V9B 06/05/75 PAGE 6

0001 FIELD 1

/CHANGE TO B,D INPUT CONVERSION ROUTINES (D.P. FPP ONLY)

6450 *16450 /'DIND'+1 - DOUBLE "D"'S SEEN

16450 5321 JMP DINERR /GO TO NEW ERROR CALL

6516 *16516 /'DINER' - ILLEGAL CHARACTER IN INPUT

16516 5321 JMP DINERR /GO TO SAFER PLACE

6521 *16521 /'DIDCPT'+2 - TWO DECIMAL POINTS SEEN

16521 4763 DINERR, JMS I AINER3 /CALL NEW ERROR ROUTINE

6563 *16563 /'DINESW'+1 - WAS FREE

16563 7400 AINER3, INERX

/CHANGE LINK WORD TO 'LPBUFE'

7372 *17372 /AT END OF 'LPBUF7' (EAE OR FPP)

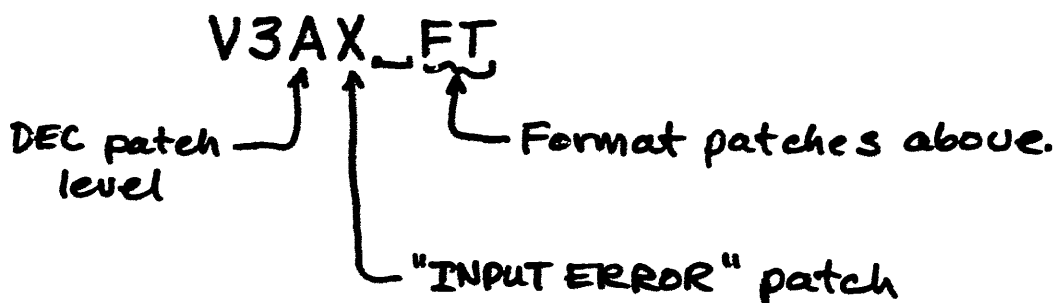
17372 7422 LPBUFE

\$\$\$\$\$\$

Change to FRTS to defeat default decimal point input and to tally ignore spaces (FRTS V3A)

2177 / 2400	2571 ↓	Change literal: address of I, G, E, F input formats.
2571 / XXXX	3064 ↓	"DCA 0D" - set default to 0
2572 / XXXX	5200 ↓	"JMP IGEFIN" - go to real start
2443 / 7640	7200 ↓	"CLA" - ignore trailing blanks
5535 / 4040	624 ↓	New version:

Version of FRTS will become:



Jim Crapuchettes
6/21/75
(done 2/15/75)

Bug fixes for FUTIL

1. Bug in access routine when in OFFSET MODE

3015 / 7750 7710 "S^A CLA" - Skip on NORMAL or FILE mode

2. Bug in double add routine causing early termination of WORD and STRING searches.

5451 / 1031	7100 ↓	"CLL"
05452 \	1025 1031 ↓	"TAD OPER1"
05453 \	3025 1025 ↓	"TAD ACC1"
05454 \	5032 3025 ↓	"DCA ACC1"
05455 \	7430 7004 ↓	"RAL"
05456 \	7001 1032 ↓	"TAD OPER2"

Jim Czapuchettes
6/20/75

C DEMO OF MIXED FORTRAN/SABR CODING, ALLOWED IN
 C OS-8. THIS LETS US DO PRIMES THAT ARE "REAL"
 C (FLOATING NOS.) AND LARGER THAN 2047.

```

S      JMP *TAG          /JUMP TO START OF PROGRAM
S      CPAGE 17         /MAKE SURE THERE'S ROOM FOR FOLLOWING:
S      LINEFD, 212
S      MINUS6, -6
S      COUNT, 0
S      EJECT, 0         /SUBR. FOR QUICK VERTICAL TAB
S      TAD MINUS6
S      DCA COUNT
S      TAD LINEFD
S      LOOP1, TLS       /TYPE IT
S      WAIT, TSP        /DONE YET?
S      JMP WAIT         /NO
S      ISZ COUNT        /YES, DONE 6 YET?
S      JMP LOOP1        /NO
S      CLA CLL          /YES, CLEAN OUT 212
S      JMP I EJECT      /HARDWARE RETURN TO MAIN PROG.
S                          /GENERATES 2 INSTRUCTIONS

```

C THE ABOVE IS A PURELY LOCAL SUBROUTINE
 C ALL OF IT CAN BE CUT OUT, EVEN "JMP *TAG", IF
 C YOU AREN'T INTERESTED IN "EJECT". IF SO,
 C THEN THE TAG "*TAG" CAN GO, TOO. CF. BELOW:

```

S      *TAG, CLA CLL
S      TAD (2330        /DENORMALIZE "ZERO"
S      DCA \ZERO        /1ST WORD OF "ZERO"

```

C BACK TO FORTRAN:

```

C DO A CR/LF TO START:
90 WRITE (1,350)
   KOUNTR=0
S           JMS EJECT           /6 LINE FEEDS
1 WRITE (1,100),ZERO,
C "ZERO" IS A FAKE WRITE--SEE FORMAT
   READ (1,350),BOTH
   IF (BOTH - 5.) 1,2,2
2 WRITE (1,200),ZERO,
   READ (1,350),TOP
   IF (5.E6 - TOP) 2,3,3
3 CONTINUE
S           JMS EJECT           /6 LINE FEEDS
C SHENANIGANS TO MAKE "BOTH" AN ODD NUMBER:
   BOTH = BOTH/2. + ZERO
   BOTH = 2.*BOTH + 1.
C END OF SHENANIGANS. INITIALIZE "VALUE":
   VALUE = BOTH - 2.
C ----< OUTER LOOP: >-----
10 VALUE = VALUE + 2.
   IF (VALUE - TOP) 11,11,90
11 FACTOR = 1.
C -----< INNER LOOP: >----
20 FACTOR = FACTOR + 2.
   QUOT = VALUE/FACTOR
   FIXIT = QUOT + ZERO
   IF (FIXIT - QUOT) 21,16,16
21 IF (QUOT - FACTOR) 50,16,20
C -----< LOOPS END HERE >-----

```

C ALL POSSIBLE FACTORS FAILED, SO IT'S PRIME:

```
50      KOUNTR=KOUNTR+1
      WRITE (1,300),KOUNTR,VALUE
      GO TO 10
```

C INFINITE PROGRAM, TILL HALTED ('C WILL DO IT).

```
100     FORMAT ('BOTTOM VALUE, >3: ',F0.0)
200     FORMAT (' TOP VALUE,   : ',F0.0)
```

C THE SILLY F0.0 IS THE ONLY WAY TO GET IT TO TYPE TEXT
C WHEN WE INHIBIT THE CR/LF BY USING A COMMA (SEE PROGRAM LINES
C 1 AND 2, PREV. PAGE).

```
300     FORMAT (14,2X,F10.0)
350     FORMAT (F10.0)
```

END