NEWSLETTER

PS/8 - OS/8 - OS/12 - DS/8

October Number 11 1974

Contributions and correspondence should be sent to:

Bob Hassinger, Coordinator
PS/8-OS/8-OS/12 Special Interest Group
c/o DECUS
146 Main Street
Maynard, Massachusetts 01754

FINALLY!

It has been quite awhile since the last Newsletter. This was partly because there always seems to be some new piece of news about to break. This is always a good reason to put off doing the Newsletter for just a few more days. It really has been a very busy summer though. Any help and contributions will be greatly appreciated. It is getting harder and harder to keep up with everything.

OS/8 V3

As everyone should know by now, DEC is distributing the new release of OS/8 called version 3 and, as expected, it supports many new features. Also, in the release is the new OS/8 Handbook which combines documentation for all of the present system software in one place. It took a long time to get the handbook written. In fact, the software was done long before the handbook for a change. There is a new version of the Software Support Manual also. I understand that the number for it will be DEC-S8-OSSMB-A-D. I have seen a copy of it so it must be more or less available.

INSERTING SYSTEM DEVICE HANDLERS INTO OS/8 BUILD

With the new release of OS/8 there was an updated version of BUILD which is capable of accepting system device handlers as well as non-system device handlers. In fact, the system device handlers it will accept are able to have multiple entry points. Unfortunately the information in the new Software Support Manual is not really complete on how to set up a system device handler for loading into BUILD. The procedure includes the following:

1. The body of the system handler should be originated to 200 but must start with a ZBLOCK 7. The entry point must be at relative location 7. In other words, this will correspond to location 7607 when it is loaded as a system device handler.

2. The name of the system handler must be SYS: (That will be the entry point that is at relative location 7.)

3. Each handler entry point has an 8-word header block associated with it.

The following additions apply in addition to the usual information in the header block. Word 5, bits 9 to 11 should normally be zero. If the device can have multiple platters like RF-08 then this field specifies the maximum number of platters allowed. Each platter above the first bumps the internal DCB code by 1. Word 6, bit zero equals 1 means the system device is 2 pages long. The second page is origined into 400 but resides in field 2 location 7600. Bit 1 equals 1 if entry point is SYS:. Bit 2 equals 1 if this entry point is co-resident with SYS. Word 7 must be zero. Word 10, number of blocks on the device. Immediately following the header records is the code for the device bootstrap. This is preceded by minus the number of words in the bootstrap. No origins may appear in this code. It must be less than 47 locations long. This last follows all of the 8-word header blocks for the handler. The bootstrap will be located into the first 47 locations of block zero. As you know, block zero of a system device contains, in the first half, the code which gets loaded into 17600 and, in the second half, the code that is loaded into 07600, so effectively, the bootstrap lies over the area where the command decoded tables will be. The reason this is all right is because the command decoder zeroes the table each time that it uses it (this is done only in core, of course). Your bootstrap must either move what is read in from block zero into the correct locations in core or else must re-read block zero into those locations. These are essentially the same restrictions that have always existed on system device handlers.

There are certain subtileties involved in two-page system device handlers for the brave of heart who want to try to write one. A two-page system device handler has the restriction that relative location 12 - that is at execution time location 07612, must contain a 0003. This is for BUILD's sake. This is what BUILD uses to recognize the two-page handler under certain circumstances rather than looking at the bit that's specified in the header block. This works fine for DEC becauses the only two-page handler they have has that value at that location. Before you try to use this new handler with OS/8 FORTRAN IV or with OS/8 BASIC, you better find out how those programs deal with the second page in field 2. They check certain locations in the handler and if they indicate that there is a TD8e handler there with its second page in field 2, they move the handler to the top of core and they patch the CDF's and CIF's in field zero appropriately. You will have to figure out how to deal with that if you need a two-page handler.

SET COMMAND

If you have OS/8 versions 3 and look at the CCL source that comes with it you may be puzzled to notice some reference to a SET command. This is not discussed in the OS/8 Handbook. I've tracked down what the story is. It seems that code does exist to implement a set command which allows you (from the command level) to alter the characteristics of some of your handlers such as the teletype handler, the line printer handler, and the card reader handler. On large systems this is useful, for instance, where you may want to go between wide line printer paper and narrow paper, or where you might want to change from 026 to 029 card

punch code without rebuilding the system. The code to implement this command isn't presently distributed because it has been added to one of the systems programs as an extention. It was done after OS/8 version 3 was released. Because it is a systems program it's not appropriate for the modified version to be placed in the library apparently. However, if you think it is an interesting facility let me know and we'll see what we can do. Incidentally, along these same lines I have information on patching the KI8e handler (the new super teletype handler included in V3) to change its characteristics you can turn echoing off and on, alter paging, alter tab, change the filler characters, the page width, the device codes that are utilized, and you can have the ability to disable control-C from interrupting from non-console teletypes. You can turn the flagging of lower case characters on and off and you can change the lower case conversion. If people find this useful I'll consider publishing the information in the next Newsletter.

OS/8 FORTRAN IV

There is an update of the OS/8 FORTRAN IV system now in the program library. It corrects a number of minor problems and it adds a couple of new features. One is, it seems that double precision arithmetic is now supported even if you do not have an FPP-12. This was not the case in the past. It appears that the double precision input and output formats are still not supported however. The other particular feature that I've noticed was that provision is made to run FORTRAN program in the background of RTS-8. RTS-8 normally does not support interrupt programs in the background but apparently even though the FORTRAN IV run time system uses the interrupts this has been worked around.

RTS-8

The program library now has a second release of RTS-8 (note that this was previously referred to as SRT-8 until DEC chose to change the name). The new release has the particular feature of supporting swappable tasks - in other words you could have several tasks each of which operate out of the same area of core and they would all be resident on the disc and be swapped in alternately, depending on their priority, as they are required. It looks as though RTS-8 is going to be an important piece of software in the future, and we should keep our eye on it. Because it has the provision for running an OS/8 job in the background (if you have the time sharing instruction trap and enough core), it makes it possible to have a foreground-background OS/8 system. This is the only DEC supported software that can do this. I think with a little work it might not be too hard to even work up a multi-user situation in this environment.

LPTSPL

There has been several questions raised about a program called LPTSPL which is briefly referenced in the new OS/8 Handbook under the PRINT command. This is a program which, at the time that the handbook was written, existed in a rather preliminary form. It was hoped that it could be made available in some way to the users. As it turns out it could not be made a part of the supported release in OS/8 version 3 although I think that the CCL commands to utilize it

are present in the release. The plan had been to probably place the program in DECUS for anyone who needed it. It is basically a program to format a nice line printer listing of output files. You write your files on to a mass storage device and then you run LPTSPL and it prints headers and trailers around each listing the way the spooling system on a PDP-10 would with large block letters, etc. Unfortunately, while the save image of the program is available no one has been able to find a source for it so far and there are bugs in the program at the present stage. I have a copy of the save image in case anyone needs it. Either we will track down a source of the program or eventually suitable patches will be made to the save image so that it can be submitted to the library in a working form.

NEW SOFTWARE IN PROCESS FROM DEC

As mentioned in the last Newsletter, there is work going on at DEC on a new assembler for OS/8 called MACREL. This is intended to be a full-fledged MACRO assembler more or less along the lines of the PDP-11 MACRO assembler. It will produce relocatable code with all of the nice features that you'd like to have, and there will be a linking loader to go with it to do the relocating, and supposedly it will even include an overlay support the way the RT-11 linker does. Along with this there is some indication that a new FORTRAN is being worked on to go with this package. It will be something between a FORTRAN II and a FORTRAN IV, probably a subset of FORTRAN IV with added features to allow it to operate in a real-time environment. In particular, it will be useful for writing tasks for the RTS-8 environment. In fact, apparently one of the main reasons for developing this new package of software is to enhance the RTS-8 environment and make it more attractive and easier to use. At last report, work was moving along well. At the Spring DECUS Symposium we were told something about these new packages at the OS/8 Workshop. Hopefully at the Fall meeting we will be able to explore the subject further.

DECSYSTEM-8

Doug Wrege tells me that he has recently submitted an update to DECsystem-8 to the library. This is version 3. It runs with OS/8 version 3 to give most of the same extensions that the older DECsystem-8's gave although where OS/8 version 3 supports one of these features, DECsystem-8 has adopted the OS/8 version of the support. Doug still plans to implement the LOG-ON feature sometime in the future for this version and he has his 8K batch called HASP working in this version.

DECUS DEMONSTRATION OF A PROTOTYPE LIBRARY DISTRIBUTION SCHEME

Doug Wrege has indicated that he and Tom MacIntyre have been working on a prototype of a system that could run on a PDP-8 OS/8 system and which could support the type of thing that the DECUS library needs to do for distribution of programs. That is, all of the programs would reside on one or more discs and they could be requested as will referencing a master directory and outputting them on the media that is desired. This is a program that the OS/8 SIG has been encouraging because presently DECUS does not have this sort of a

capability. They are only able to duplicate programs on the same media that they are submitted on. Hopefully this kind of a demonstration will encourage further exploration of the possibilities of extending DECUS library capability in these directions. We hope to have an operating demonstration of this scheme at the upcoming DECUS meeting with some of the more interesting OS/8 library programs on the system so that people can get copies of them. If all goes well, the code gets written, we get copies of all the programs, and the right hardware shows up in San Diego, it should prove to be of great use.

SOFTWARE PROBLEMS

Be sure that you check the August 1974 Digital Software News for the PDP-8/12. There are a number of significant corrections for the newest version of OS/8 including the patch to make the CCL EDIT command work correctly.

PALOS8

I recently received a package from Jim Gosling of the University of Calvary on a program he wrote called PALOS8. It is an expanded version of PAL with certain important new features. In particular, it has the ability to assemble code into areas where it will fit. The programmer can designate sections of code that may float and at assembly time the assembler will work out how to fit all of these sections together in the best way. It also includes many new psudo-ops such as REQUST which allows you to include in your assembly a file which contains all of your commonly used sub-routines. The source code for the particular program you're working on includes REQUST's for specific sub-routines out of that package and they are assembled only as requested. Thus you can get a lot of the advantages of a relocating assembler and linking loader without having to wait for MACREL.

FOCALC

Jim Gosling has also sent along something that he has been working on that utilizes his PALOS8. It is a compiler for FOCAL. At the moment it's still at the development stage although it is working. I think he intends to improve it and it needs to be documented. It presently supports just the standard FOCAL-69 syntax. Hopefully we can explore the possibility of extending it to support some of the features of the more advanced FOCAL's.

FOCAL U/W

Jim Van Zee from the University of Washington in Seattle has sent along, and submitted to DECUS, his version of FOCAL. It is an update of OMSI PS8 FOCAL which adds numerous features to OMSI FOCAL. At the same time he has been able to keep the same amount of core available for programs and variables. He has kept the error messages the same, and in general it seems that he has done a rather fine job. His package has improved support in the floating point software for not only a standard PDP-8 but also the old EAE and the new EAE. He has managed to come up with a scheme so that you can specify input and output file names at run time rather than having to write them into the program the

way you do with OMSI FOCAL. He also has a package available which supports the CALCOMP plotter. It has the usual pen-moving ability and it also has the ability to produce lettering. You have nice controls like the ability to rotate the whole picture on the plotter at will. What is really interesting about this package and which sets it apart from previous ones is that it is available in two versions. One uses the top part of field 2 for the plotting and the other functions as an overlay within an 8K memory. To the best of my knowledge this is the first time there has been a practical way to support characters and symbols from FOCAL in 8K. This package should be available from the library before too long.

ROGALGOL

Dr. Roger Abbott has sent along an update on his version of ALGOL called ROGALGOL. He has been working on installing some of the newer, faster, floating point packages as well as adding formatted printing of floating point numbers to improve I/O.

F4EAE

I have some information from Philip Siemens from Lawrence Livermore Lab about a overlay for the OS/8 FORTRAN IV run-time system which supports the old EAE he has submitted to DECUS.

DECUS 8-143 and 8-144 (FFT-C and FFT-R)

A new copy of these has been submitted to the DECUS library on an OS/8 format DECtape so that they are easier to access. These FFT routines have been popular for a long time and now they will be more readily accessible for future work.

PLOTVS AND F4 GRAPHICS

Dennis McGhie has submitted a program he calls PLOTVS to the library. It takes a common set of plotter calls and can direct the plotting to either an incremental plotter, a Versatex Printer-Plotter, or a storage scope. In general, what he has done is to define a file format which can be interpreted by suitable sections of the code to drive these different devices. He also has a set of routines for use under OS/8 FORTRAN IV for doing plotting which he finds are very nice. He calls this package F4 GRAPHICS and it has been submitted to DECUS. The package can either drive the plotter directly or produce a file for PLOTVS.

NON-SYSTEM HANDLER FOR THE RK8e DISC

I have a note from J. D. A. Griffin at Ontario Hydro in Toronto telling about a non-system device handler for the RK8e. He has modified the standard RK8e non-system device handler to treat the top half of disc zero as four separate devices, each exactly as long as a DECtape. The SYS handler is left as is except that there are now 6743 octal blocks available instead of just 6260.

UTR7: MODIFICATION

Harold E. Cronin from the Naval Weapons Center at China Lake has written that he is submitting a modification to the UTR7: Utility Tape Reading Program to allow it's use with the TM8e magtape controller. He says that he is also working on an FOCL.S "Magtape Formatter" program modification, and also on the "7 or 9 track MTA Handler" by Roger Siemen. He plans to send along copies of them when he gets them running on his TM8e also.

TELEX 5 HOLE PAPER TAPE HANDLER

I recently received a note from Garvin Eadie in the Physics Department at Leicester University, United Kingdom, about an OS/8 handler he has written for TELEX 5 level paper tape. He says he will be submitting it to DECUS in due course. The handler uses 2 pages with 7 locations free on the first page and 12 on the second. It is totally standard, responding to control-C in the usual way, and it has no error detection. The only special requirement is that the paper tape punch has to be able to accept 5 hole tape, of course.