



Contributions and correspondence should be sent to:

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

USER INPUT TO DEC'S SOFTWARE PLANNING

DEC is finding that user input is valuable in planning what they want to do in the way of new developments. The PDP-10 Main Frame group has been leading the way in this area for some time. Now the other Special Interest Groups are coming into the picture in a bigger way.

The OS/8 SIG has been asked for inputs in our area. If DEC were to consider developing a new operating system for the PDP-8 (and so for the PDP-12 also I suppose) what should it look like? What features should it have? What trade-offs should be made between size, speed, and features? How about compatibility with the current OS/8? How much compatibility? How critical is it? How much interest would such a new system receive?

I suggested that the transition, and interest might be compared to the change from the old 4K DISK/DECTape monitor to PS/8.

A general framework for this sort of a project might be to design a system with many characteristics similar to the current OS/8, to retain as much as possible from the current cusps (PAL8, etc.) but to "merge" OS/8 with some of the RTS-8 features in the sense of supporting interrupts, larger device handlers, and providing terminal support in the monitor to free us from the perpetual problem of having the teletype support written into each program. The scope of such a project might be as much as two or three man years of development effort.

This is a great chance to influence what happens to future software development for our machines. Don't let the chance get away. Write down your ideas and send them to me. Also, come to the Spring meeting ready to talk to each other and DEC about the subject. Remember that DEC has to see enough interest to be convinced that such a project is worthwhile, needed, and economically justified.

CORRECTIONS

The last Newsletter contained two errors that have been brought to light since its publication. The first was the statement that the new version of OS/8 FORTRAN IV that's available from the Software Distribution Center includes

double precision support for non FPP-12 configurations. The statement was in error due to inadequate checking of the information on my part. It seems that the only double precision features in the software simulation of the FPP-12 remain the operations that are necessary in order to be able to support COMPLEX operations as in the previous release.

The other error contained in the last Newsletter was the statement that the second release of RTS-8 was available from the Software Distribution Center. It turns out that that information was premature and that it, in fact, the new version hasn't been officially announced as being available even now although it should be shortly if the last information I received proves to be true. This is the version that will support task swapping.

THE DECUS SPRING SYMPOSIUM

As you should know by now, the Spring DECUS Symposium will be held April 15 through 18 in Miami Beach. It appears that a large active meeting will be held. Doug Wrege has been appointed to work on the Symposium Program Committee to coordinate the OS/8 SIG part of the program. Doug has been working very hard and from the information available so far it looks like there will be an active, extensive, PDP-8 program at the meeting.

Several Special Interest Group activities will be included. There are a number of interesting papers scheduled. It's anticipated that some smaller get togethers will also be possible among people with particular interests (i.e., birds-of-a-feather sessions).

THE NEWSLETTER

As you have no doubt noticed, the Newsletter has been rather infrequent and irregular. The main reason for this is that I have to find and write most of the material. Some of the newer SIG's are publishing their newsletters on a regular basis. The reason they can manage it seems to be that the membership is submitting material to use. The DECUS office would happily put us on a regular bi-monthly or monthly publication schedule if we could get the newsletter worked up that frequently, to do it you have to participate and submit material. You can't put it off and you can't assume someone else will do it. If you don't do it now it's not going to get done.

A good idea I got from Mark Lewis regards SPR's. Mark gets his RSX-11 SIG members to send him a copy of SPR's when they submit them to DEC. Because DEC can take a long time to process an SPR and publish it most users don't hear about the problems for months after the fact. Mark publishes the information in his RSX-11 newsletter on a much shorter schedule so everyone will know about the problem, and if anyone knows how to fix it they can make the information available much quicker than through DEC's usual cycle. The key to this idea is getting you to send a copy of your SPR to me when you submit it. Even if you don't want to submit a full-fledged SPR to DEC you can still send me a description of your problem which I can publish in hopes that others will know how to solve the problem.

MACREL

The macro-relocatable assembler project is being delayed by some smaller projects (support for the new "CLASSIC" for instance) but DEC still hopes they can close up the slippage and get it ready by the original September to November time frame. A pessimist might think that the release would be delayed till after the first of the year though. Let's hope not.

No firm marketing plans are available yet. It is likely that MACREL will be offered as an extra cost item like some of the other add-ons, I think.

RTS-8 FORTRAN

The RTS-8 FORTRAN which was referenced in the last Newsletter was discussed at the Fall DECUS Symposium in the OS/8 Workshop and it was announced that that project had been put on an indefinite hold due to higher priority items. There was some disappointment expressed at the announcement as some people felt that a capability to write tasks in FORTRAN would be a great enhancement for RTS-8.

More recent discussions with various people at DEC have suggested that part of the problem may be that the design of this sort of a FORTRAN really is a problem. Virtually any higher level language implementation requires fairly extensive assembly language support in the form of a run-time system. When you get a multi-task environment there is a considerable question of how to provide run-time support for each task. On a machine like the PDP-11 re-entrant code can be written for run-time support so that one set of library routines can be linked to any number of FORTRAN modules, but on the PDP-8 it is rather difficult to do because the writing of re-entrant code is quite difficult. While this may not have been the main problem in suspending work on the project, it does seem to be influencing the reluctance to pick up the project and to complete it. I wonder if any of the users have inputs on the subject.

OS/8 VERSION 4

The first firm indications of a new release of OS/8 are starting to surface. So far it looks like a relatively small update release. Likely items for the new release include the SET command, support for DEC's floppy disk (note that the floppy disk has not been announced but several products that use it have), and a certain amount of updating particularly in the CCL area. DEC is trying to develop a single standard command language that would be implemented across product lines and software systems. Many users like the present defacto standard in OS/8, RT-11 and TOPS-10 but it looks as though a bigger, more consistent design is coming. OS/8 will probably try to be compatible with some sort of sub-set of the master plan. In any case, you can look for some additions to give better compatibility between OS/8, RT-11, and TOPS-10. For example, an "=" followed by a non-numeric argument will function the same way as back arrow and left angle bracket due to separate input and output file specifications. Also a ":" that follows a switch (i.e., "/E") will work the way "=" works now to specify a numeric argument. It looks possible to make these extensions and still retain compatibility with the old (current) way of doing things. The advantage will be a higher level of compatibility across systems for those who need it.

Among the other potential extensions for the next release is an "intelligent" handler called "DUMP:". It is an output device handler but it processes the data and generates formatted output on the line printer. The idea is to be able to dump a formatted listing of a binary output file being produced by an assembler or compiler without going through a special program. The teletype can be used to control the actions taken by the handler with several options provided.

The new version of OS/8 might be available in the same time frame as MACREL if it does not grow into too big a project, but I don't think any plans are definite yet due to the need to get MACREL out as near to schedule as possible.

OS/8 SIG ACTIVITIES

One of the subjects that came up at the Fall DECUS Symposium was the perpetual discussion of how to improve the quality of DECUS library content. An approach that has been proposed and that we have been trying to pursue is the idea that, while we don't want to exclude any contribution from the library as a matter of policy, we could flag, or feature programs that had been tested and approved by some mechanism. This would point potential users to many of the useful and interesting items in the library which a group of knowledgeable users have found to be particularly worthwhile.

The approach that is being pursued is to try to get the various Special Interest Groups to organize this sort of a review process in their separate areas. The RSX-11 group has already started although the volume of programs they have to deal with is relatively small so far. It seems that the thing that needs to be done first is to establish review criteria that can be used by program reviewers. While the most desirable sort of a review would involve a critique of the approach, content, etc., of a program, experience has shown that the first step, and the one that can be done effectively, is an objective review. This would involve such basics as - Is all of the material available? Will the program assemble? Will it run on various configurations? and all of that level of basic review. It seems that a lot of programs can't even pass this elementary test for many reasons. An example is a program that recently came to my attention. I made a trial assembly of it and discovered that it gave assembly errors. The reason was that it had been written and assembled under an older version of PAL8, apparently. It contained a symbol named RELOC. When I tried to do the assembly under the now current version of PAL8 errors resulted due to the fact that RELOC is now a pseudo-op for the assembler. This sort of problem has shown up in the PDP-10 library with increasing frequency as the PDP-10 software advances. Recently the sort of review that I am suggesting has been undertaken in the PDP-10 library under the auspices of the PDP-10 Main Frame Group. They have identified programs that could not be assembled, etc., and separated them into a separate category in the library, and retained the still useful material in the principle part of the library.

What the OS/8 Special Interest Group needs to pursue this very valuable project is a set of possible review standards for us to debate and adopt. I would like to receive as many suggestions as possible so that they can be combined and put

out to the membership for comments. Hopefully, the standards would be somewhat in the form of a set of points that a reviewer would be able to check. You should propose the point be checked and indicate the way to check it. Please think about this and send along your suggestions as soon as possible. You may want to bring them along if you are coming to the Spring meeting. Hopefully we can talk about ideas at that time and make a lot of progress.

USERS SUMMARY CARD

On several occasions it's been suggested to DEC that a new pocket reference card for OS/8, similar to the one that was published some time ago, would be useful. At the moment DEC does not seem ready to publish one however.

The idea came up at the Fall meeting that perhaps the users could put together their own card and make it available. We have at least two different ways to get the card printed if it comes to actually printing a card. What we need is some volunteers to help prepare the material for such a card. An alternative to actually printing a card would be to produce a machine readable file that could be run off to produce printed summary forms. The file could be formatted to a size that would fit on file cards, for instance. John Alderman has a system of this sort that he uses where he keeps a small file box near the computer with file cards for each of the system software components. The advantage that this sort of an approach has is that you avoid a sizeable investment in printing that would be wiped out by a new release of the system. Instead new features can be added and documented in the file for very little cost. Also, this sort of a file can be circulated on an individual basis and through the DECUS library quite simply. We need volunteers to work on this project. Write me if you are interested.

LOCAL USERS GROUPS

There seems to be a growing trend in DECUS to form Local User Groups within the framework of the Special Interest Groups. An example of this is the many Local User Groups within the PDP-10 main frame group, and also several LUG's have been formed or are forming in other areas such as RSX-11. The first activity of this sort in our area has recently come to light in the forming of a Swedish OS/8 Users Group. Torbjorn Alm has written to tell me that the group was recently established. They plan to meet at least once a year to get the Swedish users of OS/8 together and they are getting together their own collection, or compendium, of non-DECUS programs among the various members. At their meeting held in January they had 20 users present which is quite a good turnout for a typical Local Users Group meeting. I hope that this sort of thing will increase. If you need help forming a group of this sort let me know. The DECUS office and I can help in several ways.

PROBLEMS WITH MULTIPLE OUTPUT FILES

I will attach a memo written by Stan Rabinowitz that documents in more detail a class of problems that can arise when you write a program that uses multiple output files. The problem can come up in particular when you have a program that does multiple passes on the input with different output files. An example would be a program like PA18.

SET COMMAND

The SET command has been mentioned several times. It will allow modifying the actions of various handlers in the system. For example, the width of the paper that the line printer handler assumes can be changed without re-assembling and re-building the system. While we are waiting for the SET command to become available I will publish a memo Stan Rabinowitz wrote that documents the things it does in the case of the new KI8e teletype handler. You may find the information useful if you need to make such changes from time to time. You might want to write your own program or use something like FUTIL to make the changes you need.

LPTSPL AND DS/8 UPDATE

I have a note from Clyde Roby indicating that he has moved from West Virginia University to Georgia Tech where he is now working with Doug Wrege. They will be presenting some papers at the Spring DECUS meeting on DECsystem-8 Version 3. In particular, they have now implemented the LOGON and KJOB feature in the new DECsystem-8 and they are doing work on an improved DIRECTORY command. Also, Clyde indicated at the Fall meeting that he would be willing to solve the problem mentioned in last Newsletter about LPTSPL. He has some code that can be molded into a LPTSPL program. Clyde has promised to try to have a working version of the program for the Spring meeting. LPTSPL you will recall is a program which prints file names in large block letters on your output listing, etc., the way it is done on big systems like the PDP-10.

WEST VIRGINIA UNIVERSITY NEWS

I recently talked to Tom McIntyre at West Virginia University about the work that he's doing. He indicated that he has submitted his utility package of various useful OS/8 routines to the library and he indicated that he is planning to submit a sub-set of his file maintenance package. These are the programs he calls MEDIAI and FIND. They create a master index of all of your tapes and other storage devices and then search the master index to find where a program is stored for you.

CORE:

Apparently a growing number of OS/8 users are upgrading their hardware to a 32K configuration. One of the reasons for this is the same one that I had when I did it. That is, that there is at least one package on the market which makes this particular upgrade from 8K to 32K very attractive (about \$5000). It will soon be possible to make this the same upgrade with several different products I suspect. DEC's memory prices are coming down rapidly too.

This is potentially very good for OS/8 users in particular. It offers an unusual opportunity for those people who have a tape only system. These would be PDP-12 users who don't have a disk but who do have LINCtape and people like myself who have a PDP-8 with only DECTape and no disk. We spend a large part of our lives waiting for the tapes to spin back and forth to the system area

and to the directories on the tapes. Because of this, shortly after I got my extended memory I wrote a little routine to utilize 16K of the memory in order to speed up the operations of the system. The upper 16K of memory holds the entire system area of the system tape, as well as the directories of both SYS:(DTA0:) and DSK:(DTA1:). In spite of the fact that my code has bugs in it and it's not that easy to use, it has enabled a great improvement on my system. Many system related functions can be done nearly as fast as with a disk now instead of spending long periods of time waiting for the tapes to spin back and forth. I haven't submitted this code to the library because I felt that it was of very limited interest.

Professor Randall from Indiana University recently got the same type of memory expansion for his PDP-12 and he immediately wrote a more comprehensive package for his system. He has sent along a short description of what his package does and I will include it in this Newsletter. His material has been submitted to the library on a LINCtape as it is specifically for a LINCtape system. The handlers for the tape are specifically LINCtape drivers. It would not be very hard though, I think, to substitute DECTape handlers for his LINCtape handlers however. He has put together a package of routines and handlers that pretty well cover what you might want to do in this area.

SD8SY and SD8X

You may recall sometime ago I suggested that it might be possible to write a TD8e DECTape handler that would allow the interrupts to be turned on some of the time. Wilhelm Vandermark has done just that. He wrote a system device handler and a non-system device handler for the TD8e DECTape, each of which can be used with the interrupts enabled. They will disable the interrupts only while the tape blocks are actually being read. The interrupts are enabled between blocks so that devices such as a moderate speed terminal can be serviced quite nicely. These handlers would be of interest to people using CS/8 FORTRAN IV who would like to be able to leave the interrupts on as much of the time as possible, even while doing I/O to the tapes. They would be able to keep their line printer or terminal, or what have you running, while the tapes were in operation instead of having to have the interrupts disabled all during the use of the TD8e tapes.

Another area where this handler could prove to be of considerable value would be to people using RTS-8. They might like to take one of these handlers and modify it to be used in the RTS-8 environment. Recognizing that there are restrictions such as the interrupts being turned off for the duration of a 129 word DECTape block it still might give considerable utility in certain applications.

TECO WITH THE VC8e DISPLAY

Bob Strachan has sent along an overlay he wrote for TECO Version 3 which uses the VC8e point plot display, to display what TECO is doing more or less the way the PDP-12 uses its scope when running TECO. Apparently Bob wrote this without having a listing of TECO which is a pretty good trick. He also has sent along an EAE simulator which originally intended to work with Doug Wrege's SPACEWAR

package. I'm not sure how and when this package will reach the library but in the meantime I have a copy and you can contact either me or Bob for further information.

IBM 2741 SUPPORT

At the Fall meeting John Alderman repeated his offer of access to his IBM 2741 terminal to print essentially camera-ready copy of machine readable files for papers submitted for the proceedings and for other purposes. An IBM 2741 is essentially a Selectric typewriter hooked up so that the computer can drive it. They are often used in time-sharing systems, particularly the IBM type systems, and like all Selectrics they produce very nice looking copy compared to say a teletype or most line printers.

In connection with John's use of 2741, he has written a 2-page output handler to drive it. The reason that so much code is needed is that the 2741 uses a different code from ASCII and so a large translation table is needed to do the code conversion. If you are interested in John's offer contact either him or myself for further information. John's handler is written specifically for his interface but it could be adapted easily to similar interfaces. Let me know if you are interested in this item and I will see if John will make it available.

WORDOP.RA

At every DECUS meeting the laboratory users of OS/8 FORTRAN IV lament the fact that there seems to be no function for directly accessing 12-bit words of data under OS/8 FORTRAN IV. All data types are 3 word floating point (although INTEGER, LOGICAL, etc., data types are synthesized using this data type). Like everyone else who does certain types of laboratory and other work, I needed a way to fetch and store 12-bit words and convert them into data types that FORTRAN could work on, and I discovered that indeed a pair of routines do exist called WGET and WPUT. They are used exactly the same way as the CGET and CPUT routines that are in the standard library only they work on 12-bit data instead of 6-bit data. I found them published in the TSAR Manual. I have extracted them and made them into a separate routine. If anyone is interested in them, let me know. If there's enough interest I might see if they could be submitted to the library.

IFORM AND FFORM

Sometime ago I received a note from Dennis McGee about one of his plotting routines that I may not have mentioned in the Newsletter. They are a pair of short FORTRAN IV sub-routines which facilitate numeric plotting. They are roughly compatible with FORTRAN I and F formatted output. He sent me copies but I'm not sure whether he ever submitted them to the library. If you are interested you can let me know.

PROGRAMS IN DECUS

As those of you who have read the latest DECUS library catalog update know there are a number of new items in the program library. Some of them are:

- DECUS 8-724 Computer Catalog System. This is a set of programs written in OS/8 FORTRAN II and SABR which allow an unsophisticated user to set up a data file in which he saves information on vendor's catalog contents, categories, etc., and then he can do inquiries into file to extract various categories, etc. for reference. It's an interesting application of OS/8 FORTRAN.
- DECUS 8-726 An OS/8 Handler for the Varian STATOS 21 Line Printer. As the name suggests this is a 2 page OS/8 compatible handler to drive the Varian STATOS 21 electrostatic line printer.
- DECUS 8-731 MEMO IV. This is the latest in the long history of Gregory Ruth's program called MEMO. You may recall that in the past the DECUS library has contained MEMO, MEMO II, and MEMO III which was a modification done by Dr. Lewis that was later removed from the library because Dr. Lewis could no longer support his version as he had moved on to the PDP-11 world. The original author has submitted yet another version which is OS/8 compatible as well as compatible with his earlier versions and appears to have some improved features. MEMO IV is a program written for OS/8 that produces left and right justified page text from free form text. The intention is to permit the user to produce a readable and neatly formatted document with minimal effort. As such it seems to be a nice, simple package. Its capabilities are somewhat like the famous DEC program called RUNOFF. There have been versions of RUNOFF kicking around for awhile now written by people at DEC. It's possible that eventually it will be a supported product although there's no firm indication of that at this time. In the meantime, Tom McIntyre at West Virginia University has a program that was originally worked on by Clyde Roby that he has offered for sale. The latest version appears to be a super-set of RUNOFF. If you can afford to buy the software their program has a great many capabilities, and if you can't afford it MEMO will give you the basic capabilities. I just started using this type of program recently to help in preparing documentation for a program I'd written and I found it quite useful and very convenient. The sort of documentation I'm thinking of can change a good bit and get edited frequently. These programs save repeated retyping and give the person preparing the documentation more control over how it is laid out and presented.
- DECUS 8-732 BAVIRF A Virtual File UDEF for OS/8 BASIC. This is a set of user defined functions for use with OS/8 BASIC. They permit random access to data in up to four numerical files which may be fixed or variable length. Two functions will retrieve or deposit values

into any location of a file. Variable files are automatically expanded as needed. Users may switch from random to sequential access and vice versa. This could be a very useful package for OS/8 BASIC users wishing to access relatively large data files, particularly if they used random access.

- DECUS 8-734 Micro-Processor Language Assembler for OS/8. This is a program written in PAL8 which is a modified version of MLA the cross-assembler for DEC's micro-processor based on the INTEL 8008 chip. This is the obvious extension of DEC's assembler for the micro-processor to the OS/8 environment. The basic assembler is written to work with paper tape, I believe, but of course anyone with an OS/8 system would never tolerate that sort of thing if he could use his OS/8 files - that's what this program is for.
- DECUS 8-735 DSP8 Diagnostic Support Package for the PDP-8. This is the package of code that John Alderman has reported on at previous DECUS meetings. It is a collection of useful sub-routines and conventions for programming a small computer, in particular the PDP-8 family. They are specifically designed to facilitate the task of the diagnostic programmer in creating diagnostics to test hardware peripherals. While John intends this package for diagnostic programmers the package includes a number of generally interesting and useful routines that other programmers might like to have access to.
- DECUS 8-719 OS/8 Software for a TC58 MAGtape Control. This is a package of three programs which extend OS/8 input-output capabilities to include the TC58 magnetic tape. The first program is a TC58 device handler (2 page) that includes six special function calls and can use any desired tape recording format. The second is a set of nine SABR routines (FORTRAN callable) that provide formatted and unformatted tape input and output and special functions such as end file, spacing forward and reverse, and rewinding. The third is a SABR main program which allows the operator to position and write end of file marks on a tape, dump tape records in octal and write test data on tape.

LAST MINUTE ADDITION

TECO MACRO LIBRARY PACKAGE

I have just received a package of material from Kenneth Maxfield that illustrates a TECO package he is developing which implements an extensive macro library scheme for TECO. If you wanted to use the macro called TAB to do a conversion between spaces and tabs in the file FOO.PA you would type something like .MUNG L,TAB: FOO.PA:SS. (The "SS" indicates SYS: for input and output). This is the first work I have seen that starts to realize the potential TECO and the MUNG command. Present plans that Mr. Maxfield will run a session on this topic at the Spring meeting.

Note from Jim VanZee

Jim has sent a patch for his U/W FOCAL to fix a serious problem in the ZVR (Zero Variable Feature). The bug has been documented by Jim Cropuchettes in both U/W FOCAL and OMSI P/FOCAL.

Jim has a new version of his TEXT (readable punch) handler to make it compatible with the new OS/8 V3 handler calling conventions (this is identified as Version B). Jim is also working on an extended version of his U/W FOCAL.

More details to follow in next newsletter.

OS/8 Device Handlers for PDP-12 Core

PDP/12 users who have LINCtape as their only mass storage device are well aware that OS/8 spends a major part of its time shuffling directories, device handlers, USR, etc., between 8K of core and the first few blocks of the system tape. Many PDP/12 configurations have recently gone to 32K opening up the possibility of using the upper part of core as a rapid-access storage device. Device handlers have been written in which all or part of the upper 24K can be used as a directory device or in combination with tape (HYBRID mode) to reduce tape shuffling significantly. This note gives the plan of how these operate and gives some of the problems encountered in implementing them on OS/8 version 3. The handlers, initialization routines, and documentation are being submitted to DECUS.

A one-page non-system handler can be written which treats the top 24K of core as a 96-block directory device. The block number given in the calling routine is used to set the data field and first address; the array must be able to straddle data fields. This configuration is particularly useful as the working area of an editor where one may want to move through several blocks looking for character strings. It can be called with a version of BUILD which has the core limited to 8K and the handler active. PIP may be used to zero the directory initially and to move files in or out.

A one-page system handler can be written which uses the top 24K of upper memory as a 96-block system device. Since the system uses the first 56 blocks, this leaves only 40 for files. However, this operating system is very responsive to keyboard commands and quickly accesses directories and some files on the two LINCtapes as non-system devices. Initialization can be achieved by R. CORESY, a routine which moves the system directory blocks 7-67 into core; zero's the directory; and chains to a version of BUILD which has the handler active and core limited to 8K. Note that a full 32K system can not be rebuilt from a 8K system without changing the core limitation stored in 07777 with ODT.

HYBRID handlers have the option of using tape for the devices, or upon changing a switch in the handlers, using upper core for the first part of the device and tape for the last part. For the hybrid system device directory blocks 0-67 are located in core at 40000-73777 and blocks 70 and beyond come from tape unit 0. This means that the tape will move from file to file without intervening returns to the first part of tape; movement times are determined by distances between chained files, not by their distances from the first of the tape. Frequently-called routines, such as PIP, can be given multiple names and placed at the first and last of the device.

Although called two-page handlers, the first page of the system handler has less available space than the second. With the handler switch set for hybrid, directory blocks 0-67 in the calling routine are handled on the first page and use core as the device. Blocks greater than 67, and all blocks for tape-only mode, are handled on the second page. To initialize this, block numbers (BN) 0-67 must be moved from tape to core and the switch set for hybrid. When done the switch must be returned to tape setting and the contents of core moved back to tape. These operations can be done with simple routines so that all the user has to do is to .R HYSY to start and .R TAPESY to exit.

- 2 -

Two-page system handlers have special quirks. Some routines expect to find a 3 at 07612 in the handler. BUILD puts in up to 47 words of a bootstrap loader at block 0 but does not update the device residency table which some bootstraps move from BN 0 into location 17647-17777. Instead, the correct information for these locations is in the first half of directory BN 66 (tape block 154) and the handler page two is in the second half of BN 66 (tape block 155). Therefore, the boot starting in LINC mode in locations 4000-4046 must read tape block 1 into 07600-07777 (the first page of the handler); tape BN 154 into 17600; and tape BN 155 into 27600 (the second page of the handler).

A two-page non-system HYBRID handler can use upper core (74000-76777) to keep the LINCtape unit 1 directory always in core so that tape moves from file to file without returning to blocks 1-6. When this is combined with the HYBRID system tape operation, optimum timing is achieved; and the bottom 16K is still available for programs. The non-system hybrid handler has a switch which is set for tape or tape/core modes. The hybrid use must be initialized by a .R HYL1 which uses USR to find the device BN, reads in the handler, moves LINCtape unit 1 directory into core, sets the switch for hybrid use, and returns the device handler to the system area. When done the command .R TAPELi restores the directory onto tape and resets the handler switch to tape-only mode. PS/8 routines allowed room for only one-page handlers and will not be supported by this handler.

The major hazard of these operations is that it is easy to forget to move files and/or directories from core back onto tape before changing to another tape or quitting. Error messages have been included for some illegal combinations.

OS/8 version 3 defines the maximum available core field in location 07777 which can be set by ODT, by the CCL command CORE, or by BUILD. However, the FORTRAN IV, version 1, run-time system (FRTS) computes the maximum core and will destroy upper parts of it when using handlers. FORTRAN IV, version 2, looks at 07777 for the limits of available core; or FRTS may be modified to do this.

Besides significantly reducing editing, assembling, and compiling times the responsive keyboard commands overcome a psychological barrier which can give the operator the impression that the system is pacing him, rather than responding to him.

February 1975
James E. Randall
Dept. of Anatomy and Physiology
Indiana University
Bloomington, IN 47401

DYNAMIC MODIFICATIONS TO KL8E HANDLER

This memo assumes that some user has assembled the KL8E handler with some unknown assembly parameters and you wish to patch out certain features. This memo describes how and also lists guidelines for future source changes to the handler.

Locations referred to are relative locations in the handler. These remarks only apply to Version C of the handler (first released version) and all future source changes should strive to keep these remarks true.

1. ECHOING

To suppress echoing on input, change location 120 to a 7610 (SKP CLA). To restore echoing, change location 120 back to a 7440 (SZA).

2. PAGING

These remarks only apply to Version D. Paging refers to enabling of \uparrow S and \uparrow Q. Look at location 216. If 7650 (SNA CLA) then paging was not enabled at assembly time. If 7450 (SNA) then paging was enabled, but can be patched out by changing location 221 from a 7640 (SZA CLA) to a 7200 (CLA).

3. TABS

Search locations 200-300 for a 7. If not found, then simulated tabs were not enabled at assembly time. If found, let '7' denote the address of the location containing the 7. Then, to patch out simulated tabs (restoring hardware tabs),

- a. move C('7'-12) to location '7'-2, [i.e., set TTYTAB to be TAD TCHAR].
- b. Change location '7'+3 to a 7610 (SKP CLA).

To restore simulated tabs,

- a. Set location '7'-2 to a '7'-4&77+1200 (TAD TTY240).
- b. Change location '7'+3 to a 7640 (SZA CLA).

4. FILLER CHARACTERS

This discussion presumes that the literal 177 remains at the top of the second page.

Search locations 200-300 for a 1377. If not found, then fill characters were not enabled at assembly time. If found, denote its location by '1377'. Then, to patch out fill characters,

move C('1377'+3) to location '1377'-1 [i.e., change JMS TTYTLS to JMP PRIN+1].

To restore fill characters, move C('1377'+1) to location '1377'-1.

- 2 -

5. WIDTH

Search locations 200-377 for a 7600. Call its address '7600'.

Form '7600'+1&177+177. This is a relative location, call it T.

Then locations T and '7600'+2 contain minus the TTY width. This width must be a multiple of 10 (octal) and must be strictly less than 200.

6. DEVICE CODES

To locate TTY IOT's in the handler, just search it for any instructions of the form

6xxxy

where xx is not 20 or 21

and furthermore don't include a 6031 if 2 locations following is a 7650 (SNA CLA) in which case also don't include the 6034 2 locations prior to the 6031.

Keyboard IOT's must always be one less than teleprinter IOT's.

7. GAGING

A secondary teletype is said to be GAGED if ↑C from the console will not interrupt its activities. In Version D of the handler, teletypes may be ungaged. Feature is enabled from assembly if there occurs a 6031 which has a 7650 (SNA CLA) two locations following refer to the address of the 6031 as '6031'.

To disable (GAG) ↑C from console, set C('6031'-2) to a 7200 (CLA).

To enable ↑C from console, set C('6031'-2) to a 6034.

8. FLAGGING of lower case

Search locations 200-377 for a 247. If not found, flagging of lower case was not enabled at assembly time. If found, call its address '247'.

To disable flagging, change location '247'-2 to a 7200 (CLA). To re-enable flagging, change location '247'-2 to a 7640 (SZA CLA).

9. Lower case conversion

Search locations 200-377 for a 377. Call its address '377'. Look at location '377'+5. If this location is not a 7650 (SNA CLA) then lower case to upper case conversion was not enabled at assembly time.

To patch out feature, change loc '377'+5 to a 7610 (SKP CLA). To restore, change it back to a 7650.

Potential Problems for OS/8 Cusps
with Multiple Output Files

All good little boys and girls who read their OS/8 Software Support Manual know that you should always fetch output handlers before input handlers.

However, this technique is not good enough to avoid certain problems which may arise. This memo describes this problem which came up while writing MACREL and describes how I got around it; this technique should be useful to other people. This problem is a potential one for PAL8; the PAL8 code should be checked. Also, this info should be added to the next release of the SSM.

Consider a command decoder input line of the form

```
DEV1:,DEV2: < DEV2:,DEV3:,DEV2:
```

Now, the handler for DEV1 is loaded first into output handler area and remains there all during pass 1 while handlers for DEV2 and DEV3 swap around.

At the beginning of pass 2 we do a fetch for the handler for DEV2 and find that it is resident. If we use it, then our program crashes when DEV3's handler swaps over this handler.

The SSM tells us in such cases, we should do a RESET before fetching a new output handler. That would work, but it would do needless extra USR calls and FETCHES for simple cases like

```
DEV1:,DEV1: < DEV2:
```

or even

```
DEV1:,DEV2: < DEV2:
```

My proposed solution is as follows and minimizes the number of USR calls necessary.

FETCH output handler on each pass first before input files. However, before every call to the handler, check to see if it is still resident and get its current entry point (this can be done without a USR call); if it is not resident, then re-FETCH it.

The code to check for residency is simple:

```
TAD OUTDEV          /put out device number in AC
TAD (7646           /add in base of DHRT
DCA TEMP           /get ptr into device handler residency table
CDF 1Ø
TAD I TEMP         /get handler entry point
SNA
JMP FETCH         /it wasn't resident; go fetch it
DCA ENTRY         /it was resident; save entry point
```