PS/8 - OS/8 - OS/12 N E W S L E T T E R

JULY 1972

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

SPRING SYMPOSIUM REPORT

The DECUS Spring Symposium in Boston was well attended. The PS/8 sessions had much larger attendance than they have had in the past. This probably reflects the increasing use of PS/8. The anticipated conflict with the COS-300 (the new rework of the DIBOL system) session turned out to be no problem. It seemed that almost everyone came to the PS/8 session. One reason may have been that DEC announced that, for those who already have the hardware, COS-300 is going to cost $5000.00! Considering the reputation of its predecessor (COS-300 is a complete rework they say) it may be hard to get people interested. Unfortunately COS-300 seems to have no relation to PS/8. It uses a whole new monitor that does not have device independence, user supplied device handlers, or USR type facilities. The system only supports programming in DIBOL. You have to use special programs to convert files between PS/8 formats and COS-300 formats.

The policy on source files was restated. PS/8 system sources are still suppose to be available. OS/8 system sources (OS/8, CD, BUILD, etc.) are not available. The sources of the rest of the system programs (PAL8, FORT, etc.) will be available as with PS/8 except that the prices are higher.

During the PS/8 SIG meeting several interesting bits of information were confirmed by the DEC people who wrote PS/8 - OS/8.

1. Interrupts - it is possible to enable the interrupt at any time under PS/8 (OS/8 - OS/12) if you are sure no interrupts will be allowed from devices serviced by device handlers from the system. As it so happens, the handlers for DECtape, RK8 and RF08 mask off the interrupt. Unfortunately, the DF32 handler can not do this. If you are not using a DF32 this only leaves non-file oriented devices such as the TTY to worry about. I have heard from some people who collect data under interrupts and write it on DECtape using this approach.

2. PS/8 - OS/8 Compatibility. Because of the policy on system sources for OS/8 a number of people want to know if they can use the new versions of the programs from OS/8 under the old PS/8 monitor. The

answer is yes in all cases except ABSLDR which is an integral part of the monitor. Several people indicated their intention to stay with PS/8 due to the availability of the system source files or the cost of the new releases, and the fact that DEC announced that it intends to do up-dates on OS/8 every six months or so. This could be very expensive for us unless DEC will make a firm commitment to protect us on this point. At the DECUS meeting none of the DEC representatives would make a detailed commitment.

3. PIP. OS/8 PIP has several new features that are very nice. One is the /Y option mentioned in the last newsletter which allows moving copies of the system area from device to device and in and out of files. OS/8 PIP also seems to have corrected the problem of trying to do a /S (squish) on a system tape that is mounted on a transport other than DTAØ: (when it is SYS:). It was also stated that you can zero a system tape on other transports and retain the space for the system by combining the /Y and /Z options in one command. In the past the only way to establish a system area on a tape was to use DEC TAPE copy to copy a previously created system tape.

Little progress was made in the area of submittal standards for PS/8 - OS/8 - OS/12 programs going to DECUS and the question of DECUS developing the capability to make media conversions (so you can get a program on paper tape, for instance, even though the author submitted only a DECtape on a LINCtape). The idea was put forward that internal to DECUS they might keep everything in the form of PS/8 files on DECtape rather than the collection of paper tapes, etc., they now have. It would then be easy to pack a tape full of material on request. Unfortunately, DECUS does not have the hardware for this and Angela Cossette held out little hope of getting it in the near future.

The session on BASIC vs FOCAL was hard fought, loud and inconclusive. The languages (not implementations) are very similar. There are some implementations of FOCAL that make it extremely attractive to people with small computer configurations. There seems to be a drift towards BASIC at DEC though due to its wider availability (i.e. "standardization"). One of the principle arguments against BASIC and for FOCAL was that listings are available for the current implementations of FOCAL and it is structured in such a way that it is easy to expand and/or change its characteristics for special purposes. In general, this is not so for the many (8 or so) implementations of BASIC currently available. At the end of the session the future of FOCAL vs BASIC at DEC seemed as uncertain as ever.

Note that DEC is now officially using the name OS/12 for OS/8 when it is used on a PDP-12. There is no difference in the systems except for the selection of device handlers for LINCtape instead of DECtape.

The RK8e moving head disk was announced. It can store twice as much data as the RK8 and it costs much less. It will be an ideal system device for OS/8 as soon as DEC figures out how to address all the blocks from OS/8. They are considering treating it as two logical devices.

NEW SYSTEM SOFTWARE

DEC has released SRCCOM, BITMAP and EPIC to the program library. DEC's TECO will be released very soon. SRCCOM compares two text files and outputs the differences. This is very useful for comparing different versions of a program. John Alderman has suggested using SRCCOM to keep track of changes in the directories of a disk or tape. When he starts a session on the computer he uses PIP to write a copy of the directory in a file. At the end of the session he creates a second file of the up-dated directory and then he compares them with SRCCOM. With large directories he says this can be quite useful. BITMAP accepts a .BN file and produces an output showing which locations are used and which locations are free. The functions of EPIC are still not too clear. It seems to be a compare and edit package for binary or save files. It is suppose to be useful in up-dating system programs.

TECO is PDP-10 TECO syntax compatible and it has the added feature that if it is run on a PDP-12 it uses the scope automatically in addition to the TTY. TECO is an extremely powerful text editor with far greater capabilities than OS/8 EDIT. OMSI has reported that DEC's version of TECO runs four times faster than the one they have been offering.

At the Spring Symposium DEC confirmed that they were working on their own OS/8 BASIC and a BATCH capability for OS/8. I understand that at least 12K will be required for the BATCH, however.

OMSI has announced that they will make a new release of their PS/8 FOCAL which will include a much faster string handling capability and a means for recovering variable storage space. (FOCAL assumes a value of zero for a variable if it is not currently in variable storage so, if space is needed any variable that has a value of zero may be removed to make room for a new entry.) OMSI has a new manual which explains the basics of using TECO. I understand that the current version of PS/8 FOCAL is now available from DECUS on LINCtape for PDP-12 users.

NEW PROGRAMS IN DECUS OR NEARLY READY

Charles Moore from Rice University has submitted his second PDP-12 LINCtape of PS/8 Utility Programs. It includes a version of MAGSPY with more capabilities than the standard DIAL MAGSPY, and an INDEX program that does what the DIAL DX command does (look at and make deletions from device directories) plus a capability to transfer any or all of the files on a device. This is the fastest possible way to copy considerable numbers of files. Several other utilities including a fast LINCtape copy program are included.

Dave Kristol has submitted 8BAL. This is the macro expander program that allows you to create powerful macros. It has far more capability than MACRO-8. As a demonstration of 8BAL's power, Dave wrote a macro for doing character I/O to PS/8 devices. You include two lines of code in your program that establish the sizes and locations of all the buffers and a number of options such as XLIST for the generated code and whether or not you want the USR to

be swapped in and out. One more line of code causes the entire package of routines needed to be accessed from the macro library file and inserted in your program automatically. 8BAL will even chain to the appropriate compiler or assembler after it has done the macro expansions.

John Covert is submitting improvements to his Monitor Command Extensions package (DECUS 8-478) that give a capability to automatically delete temporary and work files from the system device after a session on the computer. Several other goodies will also be included such as a message of the day.

John is also submitting HASP. This is a package that provides a batch processing capability for people who don't have 12K or who don't want to wait for DEC to finish theirs. You create a file on SYS: that can contain as many Keyboard Monitor and Command Decoder lines as you wish. The file is automatically accessed every time an input line is needed by KM or CD. A few days ago I was able to compile, load, map, and save a dozen FORTRAN programs without intervention after I started the job. John has a little HASP file he calls CARDS that reads a stack of command cards, puts them in a command file and then starts HASP using that file. The result is a nearly full scale batch operation. HASP will work with PS/8 or DECUS 8-478. If anyone really needs to run it with OS/8 let me know.

Corrections to COMFLT (DECUS 12-70) have been submitted to correct some minor problems that have shown up.

INTIOH, the integer IOH package from FORTRAN has been submitted. It saves up to 8 pages if your program is only using integer data by eliminating the E and F formats and floating point subroutines. It also implements an L format for inputing device and file names. This works the same as the A format except that it fills out the field with zeros instead of spaces, so you do not have to type @'s to fill out the field.

INPUT (DECUS 8-480) allows free form floating point data input in FORTRAN as in FOCAL. Nice for on line data input.

Note that the most recent addition to the PDP-8 program catalog includes the various parts of the RL Monitor that were mentioned in a previous Newsletter. The Monitor itself is aimed at 4K systems with a single DECtape. For that configuration it is probably the best monitor available. PS/8 users will be interested in some of the programs like POLY SNOBOL that are designed to run under the RL Monitor but which can be modified for PS/8.

PROGRAMS CIRCULATING BUT NOT YET IN DECUS

TV14 - The latest version of the TV: device handler for the PDP-12 scope. Two page handler which now uses a pot to control the number of lines to display rather than the switch register as in previous versions.

BOOT2 - A universal bootstrap that will even bootstrap in and out of the COS-300 system.

LINC-8 Handlers - Brian Barton has written about two sets of routines he has to allow the use of PS/8 on the LINC-8. One set works on a standard machine with the limitation that the SYS: handler does not provide checksum protection. The other set gives full checksum protection but requires a minor machine modification which he has documented. If you have a non-tape system device (disk) you can use the general tape handler from the first set which gives checksum protection without the hardware modification.

FREE - A free form signed integer I/O routine for FORTRAN.

An alpha-numberic handler for the KV8/I scope that requires special hardware.

An improved version of EDIT for the PDP-12 which uses the scope and adds a WRITE command, a ZAP command that does a J without outputing the text scanned over, and a HEAD command to reset the input to the start of the input list (i.e. rewind the input).

PROGRAMS UNDER DEVELOPMENT BUT NOT YET AVAILABLE

SCOPE - The LAP6W (very much like DIAL) editor adapted to PS/8 (on the PDP-12 only). Includes ability to go back and forth through the whole file at will. Includes string searches which DIAL does not have. Also a package of useful PDP-12 programs and routines including a scope handler that uses an overlay technique.

A Relocatable Binary (SABR) Disassembler that even generate symbolic tags. This will be useful for patching SABR code when a source is not available.

SPIP - STAR PIP - A program to transfer files and delete files as standard PIP does with the added feature that you may replace the file name or the extension, or both, in the input specification with an "*". This is interpreted as meaning to accept (or reject with an option switch) all names or extensions. The effect is that if your specify

SYS: < DTA1: *.SV

you will copy all .SV files from DTA1: to SYS: Another example would be:

DTA1: < DTA2: *.*

This will copy all files from DTA2: on to DTA1: without destroying the files already on DTA1: as would happen if the /S option were used with standard PIP.

DUET - Very incomplete information so far. I think the idea is to allow a background job like listing files to go on while PS/8 runs normally. A minimum of 12K is required I think.

WISH LIST - Many items for the "wish list" were gathered at the Spring Symposium. Among them are:

DIAL to OS/8 (and visa-versa) conversion program.

LINC code assembly capability under OS/8.

FORTRAN callable versions of the new fast floating point software (both EAE and non-EAE versions).

OS/8 FORTRAN IV - Note there is a definite possibility DEC might write an FPP-12 simulator and insert it in place of the real-time portions of the existing FPP-12 FORTRAN IV.

Interrupt driven OS/8.

Multi users OS/8.

Floating point and MACRO capabilities in PAL-8 (as in MACRO-8 more or less).

Save to Binary converter (note: such software has been worked on by various users but desire is for a DEC supported version).

Upper and lower case characters (EDIT and TECO do handle both).

Multi-user real-time interrupt driven data acquisition system with high level language driving real-time operations and running OS/8 as background.

Other items that have been mentioned:

A new binary format that would make provision for identifying field setting instructions and a means of identifying external or global references while still retaining the basic form and efficiency of the standard binary generated by PAL8 as opposed to the full scale relocatable binary of SABR. This would allow much easier building of large programs from modules without the need for very large assemblies of the whole program. Another idea was to have PAL8 output a file of symbols and values that were declared GLOBAL. This file would be used as input for assemblies of other program segments.

An ability to know what field you are currently assembling in under PAL8. This allows you to write a module (or an 8BAL MACRO) that can contain memory field changes to the current field automatically.

PS/8 - OS/8 - OS/12 ON SMALL DISKS

Dr. Mark Lewis of the FAA has written about his experience with a single platter (32K) DF32 as the system device. DEC says that a larger disk is required but Dr. Lewis and others have found that the system will work with some restrictions:

1. For PAL8 programming and .SV programs, no problems are encountered unless program chaining considerations require the disk to be filled with saved images. If SABR programming is done this becomes more acute because you want to leave space for LIB8 on SYS:

2. With FORTRAN, the capacity for chaining is less of a problem than the limited space for the intermediate SABR code file produced by FORT which must go on SYS:. Dr. Lewis reports that he frequently needs to split the FORT and SABR passes because he does not have enough room for both SABR.SV and the intermediate file from the FORT pass. He also finds he must copy LIB8 on and off the disk.

In general, you should plan to do a lot of copying of files on and off of the disk because there is so little space available. You will want something beside paper tape for this because of speed and because save files cannot be put on paper tape. DECtape would be a good choice.

GENERAL I/O ROUTINES AND SYSTEM CONVENTIONS

The number of character oriented file I/O routines and the many, many variations of file I/O for FORTRAN-SABR are getting to the point where an index of them, comparing their characteristics is going to be needed. If you have anything to contribute please send it along. A comprehensive list of file name extensions in use and a list of option switches is going to be needed too. Send along anything you think should be included.

EXAMPLES AND DEMONSTRATION PROGRAMS

Mr. Tadashi Takakura of Chiba University in Japan has written to point out that there are many users such as himself who do not have ready access to anyone who can help solve problems with the system. Mr. Takakura notes that a few simple examples that I sent him easily cleared up many of his problems. He suggests that we should build up a collection of such examples to help all those users out there that don't have a DEC software specialist (who knows OS/8) available.

Mr. Takakura has sent some items to get the project started. Any contributions should be sent in reproducible quality or else as a paper tape or DECtape file that can be listed on a suitable printer.

SOME EXAMPLES OF CHAINING

FILE NAME ON A SYSTEM DEVICE		FORTRAN SOURCE PROGRAMS
MAIN	.....	MAIN PROGRAM USING CALL CHAIN
PROG1	.....	PROGRAM TO BE CHAINED

1)
```
.R  FORT
*PROG1/L
*$
.SAVE  SYS  PROG1
.R  FORT
*MAIN/G
```

2)
```
.R  FORT
*PROG1, ,   PROG1
.R  LOADER
*PROG1.RL
*$
.SAVE  SYS  PROG1.SV
.R  FORT
*MAIN/G
```

3)
```
.R  FORT
*PROG1, ,   PROG1
.R  FORT
*MAIN, ,   MAIN
.R  LOADER
*PROG1.RL
```

```
*$
.SAVE  SYS  PROG1.SV
.R  LOADER
*MAIN.RL
*$
.SAVE  SYS  MAIN.SV
.R  MAIN
```

A COMMENT ON MIXMODE OF FORTRAN AND SABR IN PS/8

Tadashi Takakura and Toyoki Kozai
College of Horticulture, Chiba University, Matsudo, Japan
College of Agriculture, University of Tokyo, Tokyo, Japan

```
#L
S       TAD (200
        I=0
        WRITE(1,10) I
10      FORMAT('   I=',I4)
        END

#E

.R FORT
*,TTY:,<TEST1/G


EXIT    0000EXT
IOH     0000EXT
MAIN    0201EXT
OPEN    0000EXT
WRITE   0000EXT
[0      0236
\I      0200
\10     0225
*A      0234

                LAP
0200  0000      \I, BLOCK 1
                DUMMY [0
                CPAGE 6
                EAP
                ENTRY MAIN
0201  7000      MAIN, NOP
0202  4033      CALL 0,OPEN
0203  0002 06
                PAUSE
                FORTR
0204  1377            TAD (200
                /       I=0
0205  3200      DCA \I
                /       WRITE(1,10) I
0206  4033      CALL 2,WRITE
0207  0203 06
0210  6201 05   ARG (1
0211  0376 01
0212  6201 05   ARG \10
0213  0225 01
0214  4033      CALL 1,IOH
0215  0104 06
0216  6201 05   ARG \I
0217  0200 01
0220  4033      CALL 1,IOH
0221  0104 06
0222  6211      ARG 0
0223  0000
```

FORTRAN compiler assumes the content of AC is zero at the end of each set of execution. Therefore, AC must be cleared in SABR before FORTRAN statements restart.

In SABR, AC is very often used as a buffer in programmed data transfer, and the content of AC need not to be stored. That is, without using DCA instruction, programs go back to the next FORTRAN statement and this might cause trouble.

The example shows that if AC is set to 200_8 in SABR before FORTRAN statement I=0, this brings erroneous result $I=128_{10}$ $(=200_8+0_8)$ This is apparent on SABR listing which tells us that FORTRAN compiler interprets I=0 as DCA \I without CLA.

A COMMENT ON MIXMODE OF FORTRAN AND SABR IN PS/8 (2)

Tadashi Takakura and Toyoki Kozai

```
                    /10     FORMAT('    I=',I4)
0224    5234        JMP .A
                    CPAGE 7
0225    5047        \10, 5047
0226    4040        4040
0227    4011        4011
0230    7547        7547
0231    5411        5411
0232    6451        6451
0233    0000        0
                    .A,
                    /       END
0234    4033        CALL 0,EXIT
0235    0005 06
0236    0000        [0, BLOCK 2
0237    0000
0376    0001
0377    0200
                    END

   I= 128
```

高倉 直

MIXED MODE OF FORTRAN IN PS/8

Tadashi Takakura
Department of Environmental Engineering
Chiba University
Matsudo, Chiba, JAPAN

An example of mixing SABR and FORTRAN statements in PS/8 is shown on page 7-16 in 8K Programming System User's Guide (DEC-P8-MEFA-D). This provides techniques how to mix SABR and FORTRAN statements. However, the following points must be clarified.

1. When arguments of FORTRAN subroutines and/or functions are used directly in SABR statements in these sub-programs, these variables must be INDIRECTLY addressed. The following example is a modified program from the example shown in the Guide.

```
      DIMENSION  M(1Ø)
          :
      J=M(1)
      DO  55  K=2,1Ø
      L=M(K)
      CALL  AND(L,J)
55    CONTINUE
          :
```

```
        SUBROUTINE  AND(L,J)
S       TAD     I   \L
S       AND     I   \J
S       DCA     I   \J
        RETURN
        END
```

2. In the same example if the statement L=M(K) is moved from the main program to the subroutine right under the subroutine statement, variable name L must be DIRECTLY addressed. That is the statement

```
S       TAD     I   \L
```

must be changed to

```
S       TAD     \L
```

3. These facts are apparent on the SABR listings.