



DECUS 12 BIT SPECIAL INTEREST GROUP  
NEWSLETTER

March

Number 33

1979

Contributions and correspondence should be sent to:

Robert Hassinger, Coordinator - 12 Bit SIG  
c/o DECUS MR2-3/E55 ..or.. Liberty Mutual Research Center  
One Iron Way 71 Frankland Road  
Marlboro, MA 01752 Hopkinton, MA 01748

DECUS/Europe contributions are solicited through:

Lars Palmer  
DECUS/Europe 12 Bit SIG Newsletter Liaison  
Hassle  
Fack  
S-431 20 MOLNDAL 1  
SWEDEN

(Please include reference to Newsletter number and page when inquiring about material published.)

NEWSLETTER SUBMISSIONS

The Newsletter is currently published bi-monthly in the odd months. The deadline for each issue is the last Friday of the preceding even numbered month. Submissions are accepted at all times and are normally used in the next issue to go to press regardless of date of receipt. The deadline for ready-to-use material for the next Newsletter is 27-April-79. Material requiring editing/re-typing should be in earlier. Ready-to-use material should use an area 6 1/2 inches (16.5 cm) wide by no more than 9 inches (23 cm) long on each page. It should be single spaced on white bond paper whenever possible and must be reasonably clean, legible and sufficiently dark for good photographic reproduction.

Material submitted in machine readable form is particularly desirable because it can be edited and incorporated into the newsletter format more easily. Higher quality reproduction is also possible this way. Contact the editor (Bob Hassinger) for further details on acceptable media and formats if you plan to make a submission in machine readable form.

SIG COMMITTEES AND WORKING GROUPS

Steering Committee:

Robert Hassinger - address above - (617) 435-3452

©1979, DECUS

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER  
Number 33 - March 1979

Jim Crapuchettes  
Menlo Computer Associates, Inc.  
P.O. Box 298  
Menlo Park, Calif. 94025  
(415) 323-3009

Lee Nichols  
E. I. Du Pont  
Experimental Station  
Building 357  
Wilmington, DE 19898  
(302) 772-3839

Jonathan Lockwood  
Harris Semiconductor  
PO Box 883  
Melbourne, FL 32901  
(305) 724-7542 M.S. 54-40

Special Steering Committee Advisors:

Tom W. McIntyre

Stan Rabinowitz

RTS-8 Working Group and US Symposia Committee Representative

Lee Nichols - see above

Micro-8 Working Group

Jonathan Lockwood - see above

COS-310 Working Group

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

ACCOUNTING SOFTWARE

Chuck Conley, the DECUS Library Director, wrote to say that the following letter from Terry Gibson is typical of a large number of inquiries he has received over the past several months. Because of the number of inquiries, he wants to encourage more dialogue along these lines.

"I have received the August 1978 PDP-8 Program Library Catalog (...). In looking through the catalog I find that the majority of the programs apply to science and other fields not related to our business, which is accounting.

"Perhaps you can give me a little guidance. We are interested in user programs that deal with accounting. I would appreciate it very much if you could tell me where to look in the catalog or perhaps send me an updated catalog that would have programs we could find useful.

"Also, I would appreciate very much knowing of the sub-groups, i.e., LUG, SIG etc., which we might find helpful to our type of business.

"I have one other question and that is, if a program is written in FORTRAN, or as a matter of fact, any language other than DIBOL (which is what we have), can we get a translated version for our equipment? Can we use only the programs that are made for

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER  
Number 33 - March 1979

the PDP-8 or can they also be adapted for our use?"

Terry is at John F. Pfaffl, C.P.A., 301 East Olive Avenue, Suite 102, Burbank, California 91502 - 848-3810, 849-3813.

It seems to me that this letter is typical of many things I see and hear, also. Traditionally, small computers were sold to people who were prepared to invest a great deal of effort to get them to do what they needed. You had to become very familiar with the machine and its software and you had to write much of your own. Maybe you even had to build your own interfaces and so on. The computer became a principal activity for someone in the organization. Typically these users were in scientific and academic situations that encourage free exchange of information and programs. This is the interest base that built the DECUS Library, Symposia and Publications activities as we know them today.

What we see now is DEC selling, directly or through third parties, to users who expect a running system, ready to go with minimum involvement on their part. This creates a problem. Typically these people seem not ready to expend the effort and resources required to understand the hardware and software to the level that allows them to answer questions such as the above but usually they are also not sophisticated enough to have been prepared to hire someone who does have the required expertise. The result is an expectation that someone is going to solve their problems without it costing them something more. Somewhere along the line this will change, people will learn they have to expect to pay consultants to help with this sort of thing and then it will become easier to find capable, qualified consultants for the job as the market opens up.

The other problem illustrated is that it seems as though most of the people who write business type applications have an outlook that is different from the traditional one regarding sharing their work. They seem to come from an environment where one expects to sell their work rather than freely sharing it in the tradition of the research and academic environments. Perhaps as a result of this difference we seldom see viable business and accounting software contributed to the DECUS library.

I wonder if others agree with this somewhat pessimistic analysis? Is the situation more hopeful than I have outlined? What can our SIG and DECUS do to serve these new users? What new or revised policies and programs would help? It seems to me that a good answer to these questions will benefit all of us. I notice that a very high percentage of all users (not just the new types) have a need and desire to do administrative, accounting, and word processing work that could fit on the traditional systems if the software tools and applications were available.

I hope our extension into these areas in the 12-BIT SIG will help. Our orientation has always been along the traditional lines of sharing and open exchange as opposed to the the more "business like" commercial orientation of some other groups. If you have thoughts on this topic, or help to offer, please communicate them to me for the Newsletter, the Steering Committee and the DECUS hierarchy.

RECENT SUBMISSIONS TO THE DECUS PROGRAM LIBRARY

The following is a summary of programs submitted to the DECUS Program Library through February 1979.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER  
Number 33 - March 1979DECUS 8-883 - Reentrant Subroutine Programming on PDP-8

"A method for reentrant subroutine programming as well as a short, generally applicable source program is presented. The method offers a virtually transparent (programmed) stack handling as well as directly addressable counters, pointers and temporary storage. The method offers a relatively easy transcription of the standard software into reentrant one."

Runs on any PDP-8 with read/write memory, uses one page in field 0 plus seven locations in page zero. In every other field (were needed) less than a half page plus 7 page zero locations are used. This seems to be a nice little package. The listing is included in the writeup and can be easily transcribed. A demonstration of how to use the package showing the conventional code for a standard subroutine and then the modified code that runs with this package is shown. Cross field calls and returns are supported and the documentation seems good. It even gives timings for the various routines used. This seems like a good package to adapt to MACREL.

DECUS 8-899 - TK4013 - Tektronix 4013 Graphics Terminal Simulator With Hard Copy

"This program performs by software, all that is done by hardware in a Tektronix 4013 storage graphics terminal with APL character set. The PDP-8 may be communicating with a host computer through teletype ports on both computers or the program may be modified to be driven by a program in the PDP-8 itself. A Tektronix type 613 (or 611 presumably) storage display monitor and a VC8E interface are required. An x-y plotter for hard copy is optional."

The only documentation in the writeup is embedded in the 100 page PAL8 listing. This listing might be of value to others writing graphics code.

BASIC8-93 - Demonstration/Games Set #6

CLOCK: Simulation of a Digital Clock on the VT05, and HANGMAN written for EDU-25.

BASIC8-107 - Age Difference Probability Calculation.

"This OS/8 (V3D) BASIC program calculates the probability, P, that two quantities are different statistically. Given a list of values and their standard deviations, P will be calculated for all possible pairs of values. Data may be input from the keyboard or from any OS/8 device. The program contains features for editing and adding data."

Seems well documented. Accepts up to 100 data items and runs in 8K. The program listing is included in the writeup. It is about three pages of code, one statement per line if you want to type it in.

FOCAL8-340 - MICFCC: Microsphere Flow Correction and Calculation Program

"Program designed to perform mathematical corrections and calculations on data obtained from radiolabeled carbonized spheres (microspheres). Tissue samples containing microspheres are placed in a gamma radiation counting system and the data punched out on paper tape. MICFCC obtains the count per minute value of each isotope in each piece of tissue and corrects for various isotope to isotope interferences plus background radiation values before calculating regional blood flow for each piece of tissue. Any number of tissue samples and isotopes may be utilized with proper program modification. To perform these calculations using a calculator would take approximately 20 times longer, often saving 10 hours or more."

Written for PS/8 FOCAL. Brief writeup includes one page listing. Modification to work with other versions of FOCAL should be quite easy.

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER  
Number 33 - March 1979

FOCAL8-341 - ASCII to E.I.A. Conversions

"This program will read an ASCII tape into the PDP-8 High Speed Reader one block at a time and simultaneously output an exact copy in E.I.A. coding. A second option is to type a Slo-Syn program on the keyboard of a model 33 teletype and have the characters punched on tape by the high speed punch. Unlimited program lengths are possible since no characters are stored in memory being immediatly punched on tape."

Written for FOCAL 5/69. One page plus listing included in writeup. No other documentation.

A NOTE FROM EARL T. ELLIS

"I recently received a RTS/8 task which will drive the TD8E Simple DECTape. The task is really just a modification of DECUS 8-671, SD8X ION Simple DECTape handler for OS/8 by W. van der Mark, which William D. Cattey (an -8 MAN-) tested with RTS8 V1 on a 32K PDP8E at MIT. The task will turn off the interrupts for at most 17 nanoseconds (PDP8E) during read/write operations only. "TD8E HANDLER FOR RTS8" has been submitted to DECUS. Persons requiring this before it can clear DECUS paperwork can write me for a copy (Paper tape, DECTape, or Floppie). I'll supply Papertape, request media otherwise. If you have software you would like others to have, I'll try to assist while it clears DECUS. I'll even submit for you, if desired."

(I wonder if Earl really means nanoseconds? - Milliseconds maybe? His address is USCG R & D Center, Avery Pt., Groton, CT 06340, phone:(203) 445-8501 Ext.296 or (FTS) 642-7274 Ext.296 - RH)

MACREL LIBRARIAN

Alistair Windram writes via Lars Palmer as follows: "I have finally started looking at the Macrel Librarian - here are my current thoughts on it:

The program will only handle LNKLIB.RB type libraries, not MACLIB.MA (at least at first) and will operate in one of three modes:

1) Listing mode

\*list-device:file<library/L  
will list the library LSD's on the given list-device:file  
(default - console or BATCH log)

Extra options:

/P list only LSD prefaces  
/S list only SECTS  
/G list only GLOBALS and ENTRY points  
/X list only external references  
(e.g. /P/S will list both subsets)  
default - /P/S/G/X

2) Split mode

\*output-device:file<library/O (for "Open")  
will separate the library into single .RB files  
(e.g. \*SLIB<LNKLIB/O will create files SLIB01.RB, SLIB02.RB, etc.)

3) Update mode

\*new-library<old-library,new-file1,new-file2,new-file2,new-file3

DECUS 12 BIT SPECIAL INTEREST GROUP NEWSLETTER  
Number 33 - March 1979

\*new-file4,new-file5,new-file6,new-file7,new-file8,etc.  
\*new-file-n\$  
will update old-library to new-library, adding new files at start  
(or as near as possible - trying to eliminate all back references)

## Options:

/E Add new files as near end as possible (same back reference note)  
/D=nnnn Delete old module number nnnn (octal) from library  
/D/N Query console for global names (e.g. SECTS) - module where name is defined  
will be deleted  
/N Query terminal for GLOBAL names to be changed  
(e.g. new-name<old-name,file-number)  
will change old-name in the given file to new-name when it is added in to  
the library (no update is done until altmode is typed)

Both the latter interactive options to be illegal under BATCH)"

'ANNOUNCING THE "HUGH" FLOPPY'

Bill Haygood sent the following:

"BIG floppies ? DOUBLE density ? Now, you can more than TRIPLE the storage on your RX01 (or RX01 compatible) floppy drive. We are offering a floppy diskette that can be read and written by the RX01 with no hardware modifications whatever; yet packs 1540 (decimal) OS/8 blocks per diskette. That's more than three times the capacity of the standard 494 block diskette. (In fact, four of these floppies nearly equal an RK05 cartridge pack!) We supply a non-system handler that reads and writes faster than any other RX01 we know of (a system handler will be available soon). If you are interested in TRIPLING your RX01 floppy storage, write or call Bill Haygood at COMPUTER METHODS, 7822 Oakledge Road, Salt Lake City, UT 84121, phone (801) 942-2300."

NOTE FROM JEFF ELMAN

Jeffrey P. Hardy wrote for Jeff Elman to say that he has some pressing questions about OS/8 FORTRAN IV V3.02 running under OS/8 V3D. He wants to find a contact in his area who might be of assistance. This is an example of how DECUS Local User Groups (LUGs) can be of value to our members. We should be working to form LUGs wherever a group of users are near enough to be in regular contact.

Jeff also writes: "My current configuration is a PDP-12 with 12K of core, TU-56 LINctape, EAE, A/D, D/A, KW12, relay register, and two Shugart SA801R floppy drives. I designed, built, and installed the Interface/Controller for the floppies, and wrote my own two page system handler which works fine (though it was hastily written, and could be improved upon greatly). I also built and installed an interface for an OVE III Voice Synthesizer, and designed, built and installed an RS-232 interface for the system console. I would be more than happy to share my designs with any interested members."

Jeff is with the Phonetics Laboratory, Department of Linguistics, C-008, University of California - San Diego, La Jolla, California 92093.

LAWRENCE H. EISENBERG  
COS-310 Working Group  
17141 Nance Street  
Encino, California 91316  
(213) 788-0354

ERRATUM

Nothing like starting out on a new career by correcting your very first mistakes. O.K., let's get on with it then.

FIRST: My telephone number was printed incorrectly in Issue 32 (Jan. 79) and the last digit (originally printed as "1") should be "4". The correct number is 788-0354. My apologies for any inconvenience. (My poor neighbor!)

SECOND: "MENU" was poorly described by me. It's features are quite excellent, especially for installations where non-technical personnel may be expected to operate the system. It is not limited to the "first" command, as I had originally stated, but is extremely versatile and can include as many batch commands as convenient for any system function or utility (although limited to one block of source program for any single definition). While the Manual is not as well documented as it should be for this fine feature, the feature makes it quite easy to retain total program control with the MENU program. (I discuss this below.)

THIRD: While not an error, DEC's Mike Dougherty has asked me to remind everyone that FLOW was not originally intended as a supported utility, but just came about due to his disdain for writing out all of those symbols in longhand. "Please," he adds, "use it, enjoy it, but don't pounce upon me with your troubles regarding it." Actually, we hope to present some features of FLOW in future issues, but this author has had no time to experiment with it as of yet.

THE WP 200 MAKES ITS DEBUT AT THE IWP SHOW IN LOS ANGELES

The WP 200 series was up and in action, with eight terminals and one printer, at the International Word Processing show in Los Angeles (Feb. 13-15). The configuration presented included two RX01 drives and four RL01s. (Also present was WPS-11 operating on a PDP-11/34, which was obviously much slower than the PDP-8 WPS 200 series systems.)

The importance attached by DEC to the presentation was underscored by the continuing presence of project manager Jack Gilmore throughout the show. Not only was he obviously exuberant over the debut of his "baby", but he also was being very protective. Also present was a very large contingent of WPS programming and sales personnel from DEC's East-Coast offices.

The WP 200 appeared to operate flawlessly (almost). Some hitches were explained away as having been "programmed out" for the purposes of the show. For example, when I entered the "CX" command (while no one was hovering over me)

the entire system crashed. (Not just the terminal at which I was operating, and not just the disk on which I was operating, but the entire system went down and had to be re-booted from scratch.) I was told that the "CX" command had been "programmed out" for the show. Somehow that explanation was not too convincing. (I noticed that WP communications protocol was utilized through modems attached to the PDP-11, and not the 200, at the show.)

Since COS-310 is supposed to operate through the CX command on the 200 series it was not possible to independently determine whether or not the DP actually will work on the system.

We were further informed, at the show, that the system on display was the only operating system in the country, and that deliveries had been moved back another month. (First delivery in April? Maybe.)

That wouldn't be the only delay, by the way. It takes just under five minutes to transfer a fairly full WPS diskette to the RL01.

#### CHANGES IN THE WIND

Nothing official, mind you, but we hear rumour that our complaint of the inconsistency between DP and WP, vis-a-vie RL01 and RX02 support may be resolved, at least with respect to RL01 support from DP. We may expect to hear more at New Orleans upon that subject. (Maybe management did put them in that locked room?)

#### PATCHES

For those of you who have received Version 8, be sure to get all of the patches (three so far), as they are necessary to avoid unwanted file transfers (during SYSGEN/B) and certain difficulties with PIP.

We found a "bug" with PIP in a command sequence. In Version 8, PIP may be controlled under a system command protocol, but it hung up in a /D (data file transfer) command on the "MORE?" prompt. Although entering a CR from the terminal resumed the protocol, still it was an obvious bug. A telephone call to Mike Dougherty resulted in a return call the following day advising that a new patch was developed and would be provided forthwith. Now, that's what I call support from DEC and dedication from one Mr. Dougherty!

#### MENU

The MENU utility is undoubtedly one of the handiest of the enhancements initiated under Version 8. MENU is a utility which permits pre-formatted information to be displayed upon the screen together with operator prompts. The prompts (identified as COMMANDS in the utility) are interpreted and operated upon as program controls or batch controls. A single prompt can utilize a batch command file of up to one block, which can include a lot of commands.

MENU may be initialized through another feature in the SYSGEN process. Under SYSGEN/B you may create and under SYSGEN/C you may delete or create automatic

"start-up", which merely means that upon initialization of the DATE command, the Monitor will perform a Batch START. (This is true if you enter a new DATE command or if you are booting the system.) Whatever you have in your START source file on the directory will be "batched". If there is no START file, then the monitor resumes command.

Generally, you would want START to read: 10 RUN MENU,cmdf1. This would commence your menu control program as the first step after entering the date command.

We have found, in the short time available, that by setting up the MENU DISPLAY function in a manner substantially identical to that of the WPS Menus, that our operators are able to switch back and forth between WPS and DP programs with little difficulty.

To maintain control, make the last instruction in each COMMAND (batch) section "RUN MENU,cmdf1", where "cmdf1" is the same or another MENU program stored on the system directory. (Of course fatal errors in running programs will return to the Monitor, and not the MENU, as with any other BATCH stream.)

#### ESCAPE ROUTINES

The use of Escape Codes can be a powerful programming feature in many cases. (See discussion regarding the subject on page 35, issue #32, January 1979.) The ESC code is decimal 27 and can be read in an ACCEPT command in the decimal position (e.g, ACCEPT (TCHAR,CHAR); the "TCHAR" is the digit response and will display 27 for any ESC sequence). Upon receipt of the ESC code, go to a subroutine to determine the next character and what, if any, action should be followed.

We use this with the GOLD MENU sequence of the WPS keyboard. The GOLD key = ESC P and, in the ensuing subroutine, the M response indicates that the program should branch to a Main Menu for that program.

The use of CTRL codes is faster, in that each CTRL code has its own decimal equivalent thereby obviating the necessity for the subroutine to accept additional characters, but the ESC [GOLD] sequences are easier for the WPS Operator to understand. [Also, many of the special function keys on the WPS keyboard are actually CTRL/\* functions, and can be utilized in programming in the same manner.]

To enter an ESCape routine under program control, the entries are:

```
100      DISPLAY (0,0,27);      ESTABLISH ESCAPE SEQUENCE
110      DISPLAY (0,0,CODE);    COMPLETE THE SEQUENCE
```

CODE may be a decimal or letter (inside quotes - ' ') entry and may be a variable. Although you can create your own meanings for most of the sequences, several do have predetermined meanings for the VT50/52 terminals:

CODE	CHARACTER	ACTION TAKEN
27	ESC	First 027 engages ESC; the second disengages the ESCape sequence.
65	A	Moves cursor up one line.
66	B	Moves cursor down one line.
67	C	Moves cursor right one position.
68	D	Moves cursor left one position.
72	H	Moves cursor to home position.
74	J	Erases from cursor position to the end-of-screen
75	K	Erases line from cursor to right margin

There are many other predetermined sequences operative upon the VT52, but you will have to experiment yourself.

As promised, we are appending a copy of the VT100 Escape Routines to this article, created and furnished through the courtesy of Michael Dougherty at the San Francisco DECUS. Note his use of the ESCape routines.

#### WE GET MAIL

While recent space limitations prevent the reproduction of all of our correspondence, we will reproduce summaries of the mail we have received and the needs of our readers. (I just received my Newsletter two days ago, and already I am receiving mail!)

From BERNARD R. REDGATE, Director of Data Processing, Framingham Public Schools, 64 Prior Drive, Framingham, Mass. 01701, telephone 877-4816: Along with several other local school systems, Framingham uses COS 300. They have found that the DATE command will not accept any date after December 31, 1979, and want to know if there is any solution.

Frankly, I do not know of any and I am unaware of any DEC patch for the date (except, of course, for a later version). However, it seems to me that with the sole exception of displaying the date in the Directory, COS's "need" for the date appears to be an exercise without a purpose. Unless the school system requires an absolutely correct directory -- with respect to dates -- I would suggest just entering earlier years and marking the media to reflect the actual dates.

If anyone has a patch for the date problem (other than with a purchase of Version 8) please let us, and Mr. Redgate, know.

All the way from Munich, West Germany: RUDI STANGE, of DIGITAL EQUIPMENT GMBH, 8000 Munchen 40, Wallensteinplatz 2, West Germany, we are informed that there are other problems with V8. In view of the serious nature of one problem described by him, I quote:

"(T)rying to move some sources from one floppy (RX1) to the system floppy via batch mode, the directory will be destroyed after 5 to 15 sources have been moved (depending on how much is already on the RX0). The old monitor [V7] did it without problems. When tried the same procedure with the system disk (RK8E) it worked fine. PIP gave similar problems on both, the RX0 and DK0 when operating it in batch mode transferring sources from DK1 or RX1 to either system disks. The problem is such, that after the 'no room' message has been displayed and the E-option has been performed (Units 1..15 undefined), a continuation of the batch process will transfer 2 to 3 more sources and then destroys the directory."

". . . [Also] how can I obtain the COS sources in order to start some work on my own? . . ."

Mr. Stange also notes that V8 appears slower, with floppies, than was V7 and uses more space. He further noted that there seems to be some work on combining the OS/8 and COS Editors to facilitate editing. Anyone know of anything being done in the U.S. (or anywhere else) on this?

With respect to the first problem, we were warned at DECUS in San Francisco, by Norm Farrington, of some problems which might be associated with PIP/E, although certainly not the one mentioned by Mr. Stange, who indicates, at least, that he does not have any data files on the diskette.

In view of the fact that PIP was completely re-written for V8, it is not unlikely that there may be some other problems, also. (The PATCHES mentioned above may involve a correction to this problem, but we have not received any information on the last two PATCHES as of this writing.)

If there are any others out there with suggestions for Rudi Stange regarding this, please let him and us know. In the meantime, we'll just ask DEC and see what we can find out.

(I wouldn't hold my breath for COS sources, Rudi!)

#### PROGRAM NOTES

Although we have not received any brilliant contributions for using displays (or Menus) from anyone out there, we had developed one which we find to be extremely helpful. By using one logical assignment, which is always open (except in those rare cases where seven assignments are needed for other purposes) and which rarely exceeds one logical unit in size, we include all of our displays within that assignment. Saves considerably on TEXT size of various programs, not to mention the programming size.

The following is a sample of the RECORD section:

```

RECORD MENU;          RECORD SECTION FOR TERMINAL DISPLAYS
  MENBR      ,D3;      DISPLAY/FORM NUMBER (REMAINS CONSTANT THROUGH
                    ;      ENTIRE DISPLAY
  MENLIN     ,D2;      ROW (LINE NUMBER ON WHICH THIS LINE OF DATA
                    ;      WILL BE DISPLAYED)
  MEN1       ,A80;     TEXT TO BE DISPLAYED (REMEMBER TO IMBED
                    ;      LEADING SPACES
RECORD
  MEN        ,D4;      DISPLAY COUNTER
  MEN2       ,D4;      TO IDENTIFY DISPLAY SEQUENCE
  MEN3       ,D4
  MEND       ,D3;      REMEMBERS FIRST MENBR (OR START OF SEQUENCE)
  MNDEX      ,100D4;   INDEX FOR UP TO 100 DISPLAYS
    
```

To create the displays, we use a simple program (WRMENU), which we run while writing a program, and which displays a "page" (24 lines) of data. (Another simple program prints out the Displays for hardcopy review.) We then fill in the lines where needed. Each line of data is stored with the row (MENLIN) and display (MENBR) numbers. If there is no data on a line, then there is nothing stored for that row, and MENLIN is incremented.

At runtime, the subroutine is told the display number (MENBR) being sought, which is saved in MEND, and it displays up to a page of data using the following:

```

PROC
  INIT (15,UP,MENU,15);  ALL DISPLAYS SAVED ON LOGICAL UNIT 15

MENU,
  DISPLAY (1,1,1);      CLEAR THE SCREEN FOR THE DISPLAY
  MEN=MNDEX(MEN3);     FIRST PROGRAM ROUTINE AFTER "PROC"
                    ;      CREATED A MENU INDEX (MNDEX)

MENU1,
  ON ERROR MENU2;      NOT LIKELY, BUT AVOID A FATAL ERROR
  READ (15,MENU,MEN)
  IF (MENLIN.EQ.77) RETURN; MENU-CREATING ROUTINE ENTERS A
                    ;      77 AS MENLIN AFTER LAST LINE OF
                    ;      DISPLAY-DATA TO DISTINGUISH END
                    ;      OF DISPLAY
  DISPLAY (MENLIN,1,MEN1)
  INCR MEN
  GOTO MENU1

MENU2,
  RETURN;              MUST HAVE READ PAST END OF FILE
    
```

We use this routine for displays of MENUs and any other information which is to be provided the terminal operator. Accept and cursor commands are entered as with any other program, but the DISPLAY materials are stored thereby avoiding many program steps and much TEXT BUFFER memory.

I hope that this information is helpful. The only way we will know, however, is if we hear from you. If the response is good, we will try to provide some interesting routines in each issue. Your assistance in providing us with some of your favorites for publication (and credit) will be appreciated.

Hope to see you in New Orleans!

*VT 100 FEATURES IN AN ORLEANS DIBOL v 8*

```

0100      START      ; SUBROUTINE 'LED'
0110      RECORD
0120 LED1,  D1          ; NON-ZERO FOR LED#1 ON
0130 LED2,  D1          ; NON-ZERO FOR LED#2 ON
0140 LED3,  D1          ; NON-ZERO FOR LED#3 ON
0150 LED4,  D1          ; NON-ZERO FOR LED#4 ON
0160      RECORD
0170 CODES,  A10,'[0; ; ; ; '
0180 P,      D2
0190
0200      PROC
0210 ; CALLING SEQUENCE
0220 ;      LED1      = NON-ZERO FOR LED#1 ON
0230 ;      LED2      = NON-ZERO FOR LED#2 ON
0240 ;      LED3      = NON-ZERO FOR LED#3 ON
0250 ;      LED4      = NON-ZERO FOR LED#4 ON
0260 ;
0270 ;      CALL LED
0280 ;
0290 ; THE SELECTED LED'S WILL REMAIN ON UNTIL 'LED' IS CALLED AGAIN. TO
0300 ; TURN ALL LED'S OFF CALL 'LED' WITH LED1, LED2, LED3, AND LED4 SET
0310 ; TO 0.
0320
0330 LED,    P=2
0340      IF(LED1.EQ.0)GO TO SKIP1          ; LED#1?
0350      P=P+2                              ; YES--
0360      CODES(P,P)=1
0370 SKIP1,  IF(LED2.EQ.0)GO TO SKIP2          ; LED#2?
0380      P=P+2                              ; YES--
0390      CODES(P,P)=2
0400 SKIP2,  IF(LED3.EQ.0)GO TO SKIP3          ; LED#3?
0410      P=P+2                              ; YES--
0420      CODES(P,P)=3
0430 SKIP3,  IF(LED4.EQ.0)GO TO SKIP4          ; LED#4?
0440      P=P+2                              ; YES--
0450      CODES(P,P)=4
0460
0470 ; ENTER ANSI MODE
0480
0490 SKIP4,  DISPLAY(0,0,27)                ; OUTPUT ESCAPE CHARACTER
0500      DISPLAY(0,0,'<')                ; OUTPUT <
0510

```

```

0520 ; TURN LEDS ON
0530
0540     DISPLAY(0,0,27)           ; OUTPUT ESCAPE CHARACTER
0550     DISPLAY(0,0,CODES(1,P))  ; OUTPUT SELECTED PARAMETERS
0560     DISPLAY(0,0,113)        ; OUTPUT LOWER CASE Q
0570
0580 ; RE-ENTER VT52 MODE
0590
0600     DISPLAY(0,0,27)           ; OUTPUT ESCAPE CHARACTER
0610     DISPLAY(0,0,'[?2']      ; OUTPUT [?2
0620     DISPLAY(0,0,108)        ; OUTPUT LOWER CASE L
0630     RETURN

```

To - OS/B V30 users of PIP V11A ( should be V12A )  
From - Alistair Winogram

V30 PIP purported to cure various problems with ZY transfers. As some of you will know, it merely aggravated the problem !! Here is a patch to PIP V11A to cure all known ZY problems.

```

13626/0000 17
13631/0000 4027
16011/3243 3240, 7001
16034/1770 1766, 3024, 5202, 4829, 2000, 0000, 0000, 5765
16044/0000 1340, 7012, 7001, 4763, 1770, 7640, 5767, 1232
16106/3243 3242
16112/3400 1600
16124/2245 2242
16134/3243 7200
16137/5240 5235
16151/7400 5600, 0000, 7770
16163/6443 6216
16167/6254 6210
16210/1377 3032, 7132, 4216, 1766, 3024, 5763
16216/6201 0000, 7012, 7001, 3207, 6201, 1607, 6211, 1232
16226/2020 7640, 5631, 5616, 6141, 0600
16622/6161 6162
16623/0100 0200

```

This patch also adds RLO1 Disk to PIP device tables, & updates patch level to V12B.

## DROP - RECOVER - COPCOM

This note describes 3 programs developed in Prof. W. L. van der Poel's laboratory at the Technical University in Delft which were distributed without documentation at the '78 European DECUS meeting in Copenhagen. The three programs: DROP, RECOVER and COPCOM, provide a method for restoring 'lost' directories or creating files from block-number specifications. They were offered on an 'as is' basis with some precautions, but even in this form, have an obvious utility.

DROP will scan an entire mass-storage device seeking clues to the nature of the files it contains. Thus if you lose the directory on a valuable tape and do not have a recent listing, DROP can be used to give you at least some idea of the files present. The program recognizes, for instance, .BN and .SV files, and in the later case, uses the CCB to suggest a likely candidate, ie PIP or ABSLDR or some other system program. Of course it does not know about any of your own special programs, but will list the starting address and the core locations used, which is often sufficient. In the case of ASCII files, DROP will print the first 60 characters (sans all Carriage Returns and Linefeeds), which again is usually enough to identify the file. Given the starting block and the length, you can then copy it with the RECOVER program.

DROP asks at the end: 'WRITE DIRECTORY?'. Respond -only- with 'NO' (or with just a carriage return). This option has not been properly implemented!

```
EX:          .R DROP
defaults:    *{output device <} {input device}
              TTY:          DSK:
```

RECOVER and COPCOM allow you to move or examine files specified by location rather than by name. Obviously when you lose the directory this is the only way to access such files, but this is also convenient for quickly copying a program to disk which is only a few blocks away from a file you have just read near the end of a tape. RECOVER uses the '=N' option to indicate both the -number- of blocks as well as the -starting block- for the transfer. Relative references can be used, i.e. 'start 10 blocks after file XXXX.PA'. The 4 low-order digits of N specify the location, and any high-order digits, the number of blocks. All numbers are in OCTAL!

```
EX:          .R RECOVER
defaults:    *{odev:filename <} {idev:}{filename}=bbbb1111
              TTY:          SYS:      (none) #blks|loc.
```

Note that you must always specify a 4-digit value for the location. If a file name is given, the location is relative to that file, otherwise not. Omitting the output device allows you to simply examine various areas before you copy them. Hopefully you have a video terminal, rather than a hard-copy device, for this purpose!

COPCOM will either -copy- or -compare- two file areas using only their location and length. (If the areas are in fact named files, OCOMP is a much better utility for comparisons, largely because of the superior output format, and then there is EPIC...) COPCOM uses the '=N' option only to specify the -number- of blocks to copy (or compare). The starting block is cleverly decoded from the 'filename' specification, i.e. 'RXA1:70' is interpreted as 'block 70 on floppy number 1' (the start of file storage on a system device). This applies to the -output specification- as well, so you can do 'block-to-block' copies if you wish (you had better know what you are doing, however!) Specifying the '/F' (file) switch returns you to the normal output filename interpretation so you can copy (as with RECOVER) any number of blocks into a named file. The '/C' switch specifies 'copy' instead of 'compare' (both begin with 'C', so you will just have to remember this one!) although it would seem that the program should be able to distinguish between these operations from the number of input 'files' specified. The comparison feature is quite useful after a directory crash when attempting to match files with those on another device. Use the '/B' option in DIRECT to get a list of locations.

```
EX:          .R COPCOM
copy         *{odev:}location < {idev:} location /C=N
copy         *{odev:}filename < {idev:} location /C/F=N
compare      *{odev:} {file} < {dev1:} loc1, {dev2:} loc2 =N
```

COPCOM has been used to copy several 'stranger' (non-DEC) floppies into OS/8 files using a byte-mode floppy handler (Newsletter #29, p15), and then comparing them with OCOMP. This would also be a convenient way to copy all or part of a PDP11 floppy to an OS/8 file, or vice versa (ibid. #30, p7).

FROM JIM VAN ZEE

## FOCAL ROUTINE OF THE MONTH

**Problem:** Calculate the number of cuts (and the size of each cut) necessary to machine off a given thickness of material, such that each cut is progressively smaller by the same amount with the last cut (LC) less than .002". Allow for a finishing pass of .0015".

**Solution:** 1.1 ZERO S,N;C TH=THICKNESS, S=SUM, N=#  
 1.2 IF (.002-LC=(TH-.0015)/S=S+N=N+1).2  
 1.3 FOR I=1,N;TYPE %6.04,CUT=(N+1-I)\*LC

**Comment:** This routine works well for total cuts of .005 to .250" The interesting logic is in line 1.2 which calculates the sum of integers (i.e. the number of times the 'last cut' is needed to achieve the total cut) from the expression:  $S=S+N=N+1$ . Each time this expression is evaluated 'N' advances by 1 while 'S' advances by 'N'. This method is so simple that it took the author -many- hours to think of it! -JvZ



National Research Council  
Canada

Conseil national de recherches  
Canada

Division of Physics

Division de physique

*File Référence*

11 Jan. 1979

Mr. R. Hassinger  
Liberty Mutual Research Center  
71 Frankland Road  
Hopkinton, Ma. 01748

Dear Bob:

The enclosed update to my previous remarks on this subject probably covers the problem more accurately - I realised that with an ordinary Teletype the lineprinter has all the time in the world to dump its buffer!

Yours sincerely,

I.M. Templeton

Ottawa, Canada  
K1A 0R6  
Telex 053-4322

MODS. TO FRTS ERROR ROUTINE  
-----

MANY FORTRAN IV USERS WITH HIGH-SPEED TERMINALS HAVE BEEN ANNOYED BY THE LOSS OF OUTPUT FROM A LONG CALCULATION WHEN AN UNEXPECTED ERROR OCCURS IN THE NEXT STEP. THE ERROR MESSAGE PRINTS AND THE PROGRAM EXITS TO MONITOR WITHOUT LEAVING TIME TO EMPTY THE LPT BUFFER. THE PATCH TO FRTS DESCRIBED BELOW SEEMS TO CLEAR UP THIS PROBLEM.

```

C      *****
C
C      PROGRAM TO TEST LPT FINISH DESPITE ERROR
C
C      WRITE(3,100)
100    FORMAT('O THIS IS A TEST LINE')
C
C      NEXT LINE CONTAINS ERROR, BUT TEST LINE PRINTS
C      IF FRTS.SV IS MODIFIED AS SHOWN BELOW
C
C      Z=SQRT(-1)
C      STOP
C      END
C
C      *****

```

A FEW LINES ARE STOLEN FROM THE LPT RING-BUFFER:

```

0003/5166 6600
0267/5166 6600
7577/5166 6600

```

THEN A SMALL PATCH WAITS FOR THE BUFFER TO EMPTY AFTER AN ERROR:

```

5052/1136 5366
5166/0000 1403
5167/0000 1004
5170/0000 7640
5171/0000 5366
5172/6600 1136
5173/XXXX 5253

```

IAN TEMPLETON  
PHYSICS DIVISION  
NATIONAL RESEARCH COUNCIL OF CANADA  
OTTAWA, CANADA K1A 0R6

PORTSMOUTH ABBEY SCHOOL  
PORTSMOUTH, R.I. 02871

January 31, 1979

SPR Center  
Administrative Services Group, SWS  
P.O. Box F  
Maynard, MA 01754

Gentlemen:

The enclosed problem in the (yes, once more!) OS-8 Handbook was brought to my attention during a visit to Canton (MA) High School today. Briefly, the program won't work (see pages 8-126, 8-127).

Two reasons, at the least, are:

A. The format width given is too narrow ('F4.2'). One must leave room for the decimal point and the minus sign -- even if it isn't there -- in field widths. The one shown allows 2 max. digits of which 2 may follow the decimal point !

B. If any FORTRAN tolerates the double-quotes shown in the FORMAT lines, it's news to me. I do not think yours does.

What is most vexing about the example is the apparent successful run shown on 8-127. This makes the user certain that it's all his fault when in fact it is not. Either someone faked the example or the program was run on a Cro-Magnon ancestor of today's F4.

Fortranopithecus Ejectus, maybe?

Yours faithfully,

(Rev. Dom) Geoffrey Chase, OSB

Slightly different, but runnable, version of the program in 8-126:

OS/8 FORTRAN IV 3.03 FEB 1 1971 PAGE ONE

C: -- the demo Fortran-IV program in OS-8 Handbook 8-126

```

0002      WRITE (4,100)
0003 100   FORMAT (1H0'THIS IS A SIMPLE EXAMPLE OF A PROGRAM')
0004      X = 2.5
0005      Y = 3.14
0006      PRODXY = X*Y
0007      WRITE (4,200)X,Y,PRODXY
0010 200   FORMAT (1H 'X =',F7.2,2X,'Y =',F7.2,2X,'X*Y =',F7.2)
0011      END

```

## "OSYBSC"

IBM compatible telecommunications package  
fully integrated into OS/8 system

A hopefully good news for the 12 bit community is the telecommunications package for OS/8 developed in the Hungarian Academy of Sciences, Budapest, Hungary.

### I. Telecommunications features

The "OSYBSC" is a software telecommunications package which allows:

1. To connect a PDP-8 to a remote IBM (or SIEMENS) mainframe. The PDP-8 emulates an IBM 2780 or IBM 3780 Remote Batch Terminal and uses IBM's standard BSC telecommunications protocol.
2. To connect a PDP-8 to a PDP-11 running an IBM 2780 Terminal emulator package (available for all major 11's op. systems from DEC), or, to an other PDP-8 running the "OSYBSC" package.

The "OSYBSC" supports different topologies of telecommunications links:

1. Point-to-point line.  
At least one end may be used by a PDP-8. On the other end there may be a mainframe, a PDP-11, a PDP-8 or an other minicomputer emulating a Remote Batch Terminal.
2. Multipoint line, with mainframe connection.  
The PDP-8 may be connected to a multidrop line and can operate as one of the slave stations of such a line. One PDP-8 can handle up to 8 logical channels. The multidrop line is connected to and driven from the mainframe.
3. Multipoint line, without mainframe connection.  
It is possible to form stand-alone star networks with PDP-8's running "OSYBSC" without requiring mainframe support. The

star is driven by a PDP-8 with the "Master Station" version of "OSYBSC". Up to 8 slave PDP-8 stations can be connected handling up to 8 logical lines each.

## II. OS/8 interface features

The "OSYBSC" package is fully integrated into OS/8 op. system. The package makes possible to use telecommunications channels from any OS/8 program which is able to handle character oriented OS/8 devices like PIP, Fortran II, Basic etc. The logical channels can be used like standard devices.

Examples:

1. Output of a file DATA.AS from DTAO: to the telecommunications line in point-to-point discipline:

```
.R PIP
*TOUT:<DTAO:DATA.AS
```

"TCUT:" is the device name for the output direction in point-to-point disciplin. The speed differences between the local devices and the telecommunications link are handled internally in the "OSYBSC".

2. Input of a remote file through the logical telecommunications channel 7 and output to line printer in multipoint:

```
.R PIP
*LPT:<T7:
```

"T7:" is the device name for channel 7 in multipoint discipline.

"OSYBSC" runs under any OS/8 V3 or later without any modifications. "OSYBSC" was tested with different FDP-11 emulator packages from DEC and with mainframes. A similar package for RTS/8 and COS 300 is also already developed.

Detailed information may be obtained from the author:

A. Szabo, Hungarian Academy of Sciences  
H-1035 Budapest III., Vörösvári ut 3.V.23.  
Hungary

Mr. Robert Hassinger  
Coordinator - 12 Bit Sig  
c/o: Liberty Mutual Research Center  
71 Franklin Road  
Hopkinton, MA 01748

Dear Bob:

This letter serves as a followup to my last letter which appeared in the "12-BIT SIG NEWS", NUMBER 31, NOVEMBER - 78, PAGE 38. I have received several inquiries concerning the OS/8 program nucleus which I have been developing. I hope the following comments will serve to explain some of the background of this software package, for those whom have not yet taken the time to write.

The software package is named "MULTIO". This acronym stands for "MULTiple Inputs and Outputs". "MULTIO" serves as a software interface between the OS/8 - OS/78 operating system and a user written, PALS assembly language program. At present, up to twenty-six (26) Mnemonic Psuedo-instructions are conditionally included in the program assembly. These provide a "High Level Language" style of input and output and other functions for use within the "User" program. The general approach of this software package is to relieve the programmer of the "technicalities" of writing OS/8 device independant programs and utilities. This allows the programmer to concentrate almost entirely on WHAT the particular program has to do, instead of concentrating on the intricate manipulations of the Operating System. For instance, initialization of the program including; calling the command decoder; converting the OS/8 system date; opening the I/O files; setting up the "run-time" options; determining the amount of memory available; and several other functions; consists of over 40 major steps, and this may be implemented with a single "Psuedo-instruction" ("INITLZ").

The most predominate feature of the package is that it provides for multiple, concurrent input and output devices to be "OPEN" simultaneously. This requirement is most often found in "SORT", "MERGE", and "DIVERGE" programs. In its present form, the package may be conditionally assembled to allow from zero to three OS/8 input devices and from zero to three OS/8 output devices to all be open at the same time. In addition, the package provides internal handlers for the console printer and keyboard, a high speed punch, a line printer, and the "BATCH" log device. This allows a user written program up to eleven (11) devices for I/O during a program run. The "internal" handlers are all selected by options passed thru the OS/8 Command Decoder. OS/8 device I/O may be either a character at a time or "Line Oriented", or a mixture of both. The "Line Oriented" I/O is implemented by High Level Language style "Mnemonics" such as "READ1", "WRITE2", etc.

The software package includes several debugging features, including selective device I/O "traps", and support for "User" program error messages. These debugging routines may be conditionally included or excluded for the final program assembly. The software package also provides for a "Trap-Trace" which shows the address of the "CALL" for each of the commonly called service routines provided in the package.

The software package also includes a large number of commonly required, "User" program callable functions, such as binary to decimal or octal conversion, push-down stack handlers, OS/8 date conversion, program initialization, character packing and unpacking and memory contents moving, device opening and closing, message printing, calling the command decoder, etc. Many of these functions are implemented by the use of "Pseudo-instruction Mnemonics" which make the resultant assembly language program resemble a high level language.

A new version of the 27-bit floating point package may also be conditionally included in the program. This package is fully compatible with the "DEC" version but runs faster and includes some new Pseudo-instructions. This segment contains several conditionally assembled sections which are used to configure the floating point package to suit the "User" program requirements.

The present PAL8 source files for this software package total 505 OS/8 blocks and the User's manual is approximately 120 pages. I intend to market the package when the User's manual is completed. It will total about 200 pages. This is entirely a spare time project but I am currently expecting to be able to release the entire package around June or July of this year. Naturally, a large user response will compel me to work harder toward an earlier completion. I will not release it prior to the completion of the manuals because I feel very strongly that it must be adequately documented. A "MACREL" version is currently being developed and will be released at a later date.

I have used this package to develop about 40 programs and OS/8 utilities to date and have implemented each of the "Upgrades" to the package so that each of the existing programs may be upgraded to the newest version by merely re-assembling the existing "User" program with the newer MULTIO sources. This has made it relatively easy for me to support a large number of assembly language programs, since the only "special" programming is the actual "User" program. This "User" program becomes a comparatively small portion of the total program (typically 5% to 45% of the total source code). The "User" programming effort is greatly reduced, since it becomes more of a "Format" or "Sequencer" for the operation of the routines contained in the "MULTIO" software package. This also causes a substantial amount of STANDARDIZATION between programs. Development effort is also reduced due to the large amount of prior testing that has been performed on the "MULTIO" software and the DEBUGGING facilities that are automatically available in this package.

Mr. Robert Hassinger

20-JAN-79

Page 3

My current "record" for an assembly language program development is for a utility program that will input an ASCII file (or concatenate up to 9 files) and output it concurrently to three OS/8 devices (Plus the line printer, high speed punch, video screen, and/or "BATCH" log, if desired). This PAL8 assembly language program was written, typed, assembled and loaded, and finally tested and debugged in less than twenty-five (25) minutes. Actually the shortest possible program that may be written with the "MULTIO" software package requires a "USER" program source that is only 7 lines long and will perform a device to device transfer of ASCII files. Including the editing of the parameter file, this should take about ten minutes to implement.

For those who wish to be notified upon the release of this software package, please send me a self-addressed, stamped envelope, to the address given below.

On another subject, I noted in one of the past issues the discussion on the possibility of transmitting paper tapes of various software programs as a part of the newsletter. The main problem of this idea was noted as being the production of the tapes and determining some means of attaching them to the newsletter. I have an alternate suggestion which may have already been discussed. Why not take some of the more useful programs that are in the DECUS library (how about the one for CCL recollections, DECUS #8-871) which are relatively short, and reproduce an assembly listing of these programs in the newsletter. Some practical limit (say 6 pages) could be placed on how large a program would be included, but this would allow those hardy souls who don't mind a little typing to type up their own source for such programs and assemble them on their systems. This would avoid the problems of media compatibility, lost or damaged paper tapes, and the sheer logistical problems of producing so many paper tapes. It may also be practical to reduce these assembly listings to half size (my XEROX machine will do this and I'll volunteer to produce the MASTER copy). This would allow more to be included in whatever the space limitations may be. Anybody who complains about the "small" print can be SHOT for looking a gift horse in the mouth. I'll work up short program and send it in for the next newsletter.

Regards,



Ken Maxfield  
11707 Macrinus  
Florissant, MO 63033

