

PEEK (65)

The Unofficial OSI Users Journal

P.O. Box 347
Owings Mills, Md. 21117
(301) 363-3267

★\$1.50★

Editor: Al Peabody
Vol. 2, No. 6, June 1981

INSIDE:

EX MON BUG FIX	3
LONG FILENAMES	5
GC BUG FIXES	8
C1P OUTPUT PROCESSOR	13

Column One

A letter to the editor this month has what was for me a fairly amazing program line in it. It says, in part,

```
D=D+7*(D>9)
```

Do you know what that line does? I didn't. Neither did Jim Sanders, at first. Then a friend of his in his office gave him a hint, and he was hooting with delighted laughter. BASIC, he explained, evaluates the expression in the parentheses first. Then, if D>9 (statement true), the expression in the parentheses is evaluated as a "1"; otherwise, it is a "0" (1). Neat for a Hex-Dec converter. But what else could you use it for? It seems to me a lot of IF..THEN statements could be replaced with these logic expressions. Best use of logic in parens submitted in time for the August issue wins a prize.

PEEK(65), like most publications, is assembled in serial, not in parallel. One fine day each month, after most of what will be printed has been submitted, edited, punched in, printed out and

pasted up on layout sheets on the wall, I spend the day with Karin and Ann, looking for typos, deciding what stays and what goes, and choosing the arrangement. Then I write this column, last thing. This means I have actually "read" the issue by the time I write these lines, so can comment on the state of PEEK(65) for the month.

Why am I telling you all of this? Because this issue impresses me. They all do, it's true, but this one in particular seems worth while, not only because we discover that Four-State Computer has practically completely disassembled 65U (Who are they, anyway, and when will they go public, and where can I get a copy???) , not only because we have the source code for a complete new polled-keyboard output processor, not only because the OSI standard communications protocol is attracting ideas and attention, but also because the ADS are so nice. Take a look at what our advertisers are offering this month. If you need something you don't see, write us a letter and someone will advertise it within a couple of months, I betcha.

After some teething problems at the new location, the PEEK(65) CBBS is on the air again at (301) 363-4867. Hook up your modem and give us a call. What teething problems, you ask? Well, the message eraser program didn't know how to keep track of the End of Data in the file, and decided to lose all the messages after the first 10 or so. Fixed that. Then the CBBS program with the new command structure lost track of the proper header for Notices, and tried to tell us there were only about 3 in the file, when I knew there were at least 17. Fixed that. Then the darn thing wouldn't answer the phone at all (modem had loose wire)... but all is working now (gee it's hard to type with my fingers crossed!).

So give us a call. And leave a letter to the editor, letting me know what you would like to see in PEEK(65). Or even leave us an article on the board, or a listing. And tell your friends.

al

THINGS LEARNED THIS MONTH

by Jim Sanders
2338 Riviera Drive
Vienna, VA 22180

LEVEL-3 HORROR STORY (part 3)

It is now for sure that the disk (which was sent by the factory to fix up the problem with Level-3 writing disk data to a file in use by another partition) is garbage. It is also for sure that the OSI people don't care. There has been no communications since the disk was sent to us, because they will not return any calls or letters. Rumor has it that they are busy working with 16-bit processors and have decided to abandon the 8-bit customers. They abandoned us long ago, but it is becoming more clear now.

The good news is that I have a verified fix as a result of a more or less complete disassembly of OS65U coming from Four-State Microcomputers. The disk will be on the way to all you folks that have requested it as soon as it is ready.

NOT YET UNDERSTOOD

Among the hundreds of mysteries that await solution are two that I have devoted a lot of hours to in the last week. Any clues will be welcomed. First, I am nearing completion on an OSU communications package and need only to figure out how to use the 128 byte ram on the CPU board. It is necessary to run a level 1 system in interrupt to continue receiving a large file while saving the buffer on floppy. To do that, you have to change the vectors for the 6502 interrupt to memory which will not be used during the program execution. With their usual foresight, OSI vectors the interrupts to 01C0 (in the stack!), but cleverly installed one of the famous

patches to pretend the 65A ROM is not really ROM. The same chip that selects the processor is able to replace the ROM with the scratchpad RAM thereby allowing you to change the vectors. But... I can't figure out how to select the RAM and the 6502 without resetting the 6502. Help?

The other is easier. Location 10287 (OSU) has the low limit character that may be stored. It used to be zero but that resulted in nulls being put on files, so it was changed to 13 (RETURN). Poking it to 07 and also poking 15314 and 1382 to 07 lets you follow REM statements with linefeeds, bells, and characters which change your printer to double width, etc. Listings using this technique can really be made to be fancy. But... there are some other locations involved that I haven't found yet. Who knows and will share?

SITTING ON IT

In addition to documenting enough of the code for OS to make your head swim, Four-State also has written a program which will recover the files and restore your disk when the power goes off or you get the dreaded ERROR 17. Another is to fix the header on hard disk lost under the same circumstances. Those of you with all those dead files on hard disk can recover them! A beauty of a program is running on their multi-user system that checks 4 bytes of memory every 40 millisecc and runs in the background full-time. It tests memory in all partitions and logs the results, ringing the system console bell when an error is found. But... they have not published yet. Send them an encouraging word to PEEK-65 and I'll forward the batch.

RAMBLE ON

The DIR fix for all versions 1.1 OSU (up) is no good for CD systems. 2AB8 should have A2/00/... etc. but if you have a CD system and tried the 'fix' you already knew that. Why doesn't the factory know that?

The recently released Random Number fix doesn't work either, unless you need numbers from zero to 0.5 instead of 1.0... but... someone has found a fix for the factory fix although I don't have it. Help?

By the way, that fix and the fix fix will prevent the DQ WP6502 Justify function from

working. Don't make a change on that disk.

The new 48K static boards have a lot of timing problems, related to the cheap timing pot and temperature sensitivity. If you use this product at 2 Mhz and find half of your program missing, switch to 1 Mhz and try the load again. Some users have had success with removing the cover to let more air in.

You can undo the fix to prevent nulls from going to a file, load a string with 127 nulls, write that string to the file repeatedly, and have a nifty 'erase' program. Good for getting rid of files you don't want to share on a swapdisk with out having the detectives resurrect your deleted programs. If you include a CHR\$(13) every 128 characters the file can be accessed with FPRINT, etc.



THE BEGINNING ASSEMBLER

PART II

It was 10 PM. I had been pounding on the computer since supper, and was getting nowhere fast. I had written, at first, a fairly complicated assembly language program, and had tried to insert it in front of a Basic program under OS-65U by using LOAD48, then call it with Y=USR(X). When it didn't run, I assumed there was something wrong with my program, so I tried to fix it... but I couldn't find a thing wrong with it. So then I replaced the complicated program with a much simpler one ... and it didn't work either.

So now, at 10, I called Jim Sanders, my favorite assembly wizard, and said something like "Help." As always, Jim helped. My program, it turned out, was fine. The whole trouble was, I didn't know how to bring it over from 65D, where the assembler lives, to 65U. Of course, none of the various manuals in my notebook tells me clearly and simply how to do this absolutely necessary process. So I will now share my newfound knowledge with you.

I assume you have doped out the intricacies of the 65D assembler, and have typed in your assembly language program, including the line "\$*=\$6000" at the beginning so that it will assemble at the beginning of the OS-65U Basic workspace. The following dialogue will get it into 65U:

Copyright ©1981 by DBMS, Inc. All Rights Reserved.

PEEK (65) is published monthly by DBMS, Inc., Owings Mills, MD 21117. Editor: Al Peabody.

Effective July 1, 1981

Subscription Rates

US (surface)	\$15
Canada & Mexico (1st class)	\$23
So. & Cen. Amer. (air)	\$35
Europe (air)	\$35
Other Foreign (air)	\$40

All subscriptions are for 1 year and are payable in advance in US Dollars.

For back issues, subscriptions, change of address or other information, write to:

PEEK (65)
P.O. Box 347
Owings Mills, MD 21117

.A3 (assembles it into RAM @ \$6000)
 .ISA 65,1=6000/B (store onto track 65 of your 65D disk)

Now insert 65U disk, boot and RUN"LOAD48, which says..

TO RETURN TO OS-65U TYPE G7E12

Now reinsert 65D disk but don't boot...

A*Z (Moves head to track zero)
 A*C6000=65,1 (Loads program from track 65 into RAM)
 A*Z (Jim says do this, so I do...)
 A*G7E12 (Returns to 65U, which types...)
 OK
 NEW 256 (MOST IMPORTANT PART ... protects the program)

Now your machine language program is in RAM at \$6000 and Basic knows not to write the first line of your new Basic program on top of it. Naturally, if your machine language program is more than 256 bytes long, type NEW XXXX, where XXXX is a number at least as large as program size in bytes.

We are not through yet. You must type in a Basic program of at least one line, and SAVE it. The program must include: POKE 8778,0:POKE8779,96:
 Y=USR(X)

That line tells Basic where to go to calculate the function USR when it is called, by inserting \$6000 in standard Lobyte, Hiyte form into the dispatch table, then calls the USR function, which is of course really your program.

Last question: how do we return to Basic? Simple. The last line of your machine language program must be an RTS. Alternately, if you LDX, Lobyte and LDA, Hiyte of any number, then JMP (8) instead of RTS, the number of which you put the low byte in X and the high byte in A will be Y. Huh? OK, if the last three lines of your assembly language program were:

LDX #5
 LDA #0
 JMP (8)

Instead of RTS, then you did the usual Y=USR(X), followed by PRINT Y, you will see a "5" appear on your screen. Why do this? No reason, unless your machine language program has several points where it might wind up, like a sort or a disk read routine which might end up with an error of several varieties. Then you could figure out where it went by loading X with a diagnostic number and jumping through 8 as above. Printing Y will then let you see where you wound up!

More next month.



A BUG IN THE EXT. MONITOR?
 Axel Unterschuetz
 DF9LX
 Bremerstr.23/907
 2300 Kiel
 West Germany

You have all heard about the checksum format, haven't you? I use it for all my machine code tapes and had put a checksum loader into the monitor PROM (\$F800-FFFF). But now after putting the Ext. Monitor into an EPROM and substituting my machine code handling routines between \$F800-FFFF by a self written BASIC support I have to rely completely on the Ext. Monitor.

There I was, loading a machine code program from a cassette when a strange thing happened, question marks began running up the screen. I rewound the tape, pressed 'space' and 'L': same result after awhile.

What had happened? The recording was bad, so a read error occurred. This should be no problem for a checksum-loader! But in this program the Load routine is aborted whenever an unexpected character (<0 or >F) comes in from the cassette. Then all following characters are interpreted as commands since the load flag is not cleared.

I discussed the problem with my friend, Rainer Willkomm DF6LW, and he found the following solution:

ADR.	OLD CODE	SUGGESTED NEW CODE
0AFF	A9 LDA#	4C JMP
0B00	3F '?'	7F
0B01	20 JSR	0F \$0F7F
0B02	61	EA NOP
0B03	08 \$0861	EA NOP
0F7F	A9 LDA#	A9 LDA#00
0F80	45 'E'	00
0F81	20 JSR	8D STA

0F82	4F	03
0F83	08 \$084F	02 \$0203
0F84	A9 LDA#	A9 LDA#
0F85	52 'R'	3F '?'
0F86	20 JSR	20 JSR
0F87	4F	61
0F88	08 \$084F	08 \$0861
0F89	20 JSR	4C JMP
0F8A	4F	09
0F8B	08 \$084F	08 \$0809

Now a question mark is put out after an error has been detected (also after a checksum error) and all reading from the cassette is stopped.

We would have preferred a real error message but could not find the space for it. Maybe one of you can. Please contact me if you have questions or suggestions.



Help Wanted

Contract work is available for

- Software Coding
- Hardware Prototyping

Send resume to be placed on the bid list

GREG PASSMORE
 9797 Meadowglen - 1608
 Houston, Texas 77042

OHIO SCIENTIFIC USERS
Business Software on Discs

Balance Sheet, P & L, Cash Flow, etc.
 Quantitative Decision Making
 Personal & Bus Financial Analysis
 Bus Forecasting Econometric Model
 Random Graphic Design Generator
 Full Manuals ("texts") & Listings

Management Education Ltd.
 Business Software Div.
 12326 Riverview
 Washington, D.C. 20022

Send Self-addressed envelope for FREE CATALOG.....



FOR ALL YOUR *OSI* NEEDS:

SYSTEMS

The complete Ohio Scientific Line.

PERIPHERALS

Printers, Diskdrives, Data Streamers,
Terminals, and more.

SUPPLIES

A full line: Diskettes, Paper, Ribbons,
Custom Forms, Data Processing Accessories.

SUPPORT

Walk-in or on-site service (Southern
California area). Software problems
corrected in hours -- not days.

SOFTWARE

Enhanced 65U and system utilities:
CPA PACKAGE -- CLIENT WRITE UP



OPUS SYSTEMS, INC.

4220 Glencoe Avenue • Marina del Rey, California 90291



Telephone: (213) 398-0966
398-6022 (24 hrs.)



LONG FILENAMES IN OS65D3

by Darrel R. "Stretch" Manley
5664 E. Evans Creek Road
Rogue River, OR 97537

I recently saw a short note (I believe in the Aardvark journal) about changing the filename length in OS65D to 14 characters. The approach was a little simplistic, since the only thing changed was the maximum length to be scanned for a name match in the operating system. I have been using 14 character file names for about 6 months, and while they are a lot nicer, it really ain't that simple, bub! You also have to decide how many files you want in the directory. The normal is 64, of course, but changing to 14 character filenames will automatically drop the maximum to 32, unless the directory is expanded and the BASIC overlays moved out of the way.

There are a few other things to watch out for. As soon as the operating system is changed, you can't do a 'RUN "FILENAME"' from BASIC unless the directory is already converted. Also, the maximum input to the operating system, while in system mode, is 18 characters. The normal operating system commands are 3 or 4 characters long, but thankfully only the first 2 need to be entered. Thus a "LOAD NEWLONGFILENAM" won't fit, but "LO NEWLONGFILENAM" will, and will be acted upon. Note that the return is counted in the 18 characters allowed. In a BASIC call to the operating system, such as 'DISK!LOAD NEWLONGFILENAM', the 18 character length can be exceeded, because BASIC points the operating system buffer pointer to either the string or string temporary in BASIC. Thus even 'DISK!LOADUPTHEFILE NEWLONGFILENAM' should and does work! The extra characters after "LO" up to the space are stepped over and ignored. Only "LOAD", "PUT", and "XQT" are affected, since they are the only operating system commands that have a filename as a parameter.

Since I have a 2 disk system (8"), all I had to do was convert one disk and put a BASIC program to convert the other directories on it. It runs from drive A and fixes whatever is in drive B.

Here are the memory locations to change for an 8" system, and what is affected by the change. All changes must be made to the disk, not as

POKES. Not all of them have to be changed, just the ones for the features you want.

Due to the arithmetic operation performed using the 2.masks, I don't advise trying less than 14 character names. It may not work correctly.

```
$2DE1 = $#07 MASK. CHANGE TO
        $#0F
$2DE3 = $#06 FILENAME LENGTH.
        CHANGE TO $#0E (14).
$2DF1 = $#F8 MASK. CHANGE TO
        $#F0
$2DF4 = $#08 FILE ENTRY
        LENGTH. CHANGE TO
        $#10 (16)
```

The above changes will allow 14 character filenames, but only a maximum of 32 files in the directory (16 each on 2 page length sectors). To regain 48 or 64 file capability, change these locations:

```
$2E07 = $#03 change to one
more than the number of
directory sectors that you
want.  $#04 = 3 sectors, or 48
files.  $#05 = 4 sectors, or
64 files.
```

```
$22AB = $#04 change to 1 more
than location $2E07, above.
This is the sector number of
the BASIC overlay for get/put.
```

```
$21A8 = $#03 change to the
same number as $2E07, above.
This is the sector number of
the BASIC overlay that
overwrites BASIC coldstart and
'RUN"BEXEC*"', on boot.
```

Now move the BASIC overlay sectors using the operating system. Move track 8, sector 4 to track 8, sector 6 (5 for 48 files). Move track 8, sector 3 to track 8, sector 5 (4 for 48 files). Convert your directory to 14 character filenames, fill the unused names and sector(s) with "#S", and you are done with the system conversion. Now change all programs that access the directory sectors directly, such as DIR, CREATE, DELETE and RENAME, to work with the new sizes.

I suggest you don't attempt this conversion unless you can figure out machine code on your own, or have a friend who can. The trained monkey approach to this conversion may get you in trouble. Also, don't do this on a disk that has something "volatile" on it. If something goes wrong, you may lose the disk contents. CAUTION!!! Don't change the name of "BEXEC" in any way. To do so will get you a "C" error on boot, and

not RUN BEXEC*.

There are a few other things to be learned in the operating system, like how to make your long programs fit in fewer disk tracks, without shortening the programs in any way; how to increase the storage capacity of files without increasing the track count; and a lot of other goodies that I'll save for another day.

All you sharpies have probably already figured out 30 character filenames from the above information. Yes, they are possible, and do work. You do lose some direct operating system input if the file name exceeds 14 characters, however.

SAMPLE DISK DIRECTORY

FILE NAME	TRACK RANGE
OS65D+	0 - 8
BEXEC*	9 - 9
CHANGE.SYSTEM	10 - 10
CREATE.FILE	11 - 12
DELETE.FILE	13 - 13
RENAME.FILE	14 - 14
BASIC.TRACE	15 - 15
ZERO.FILE	16 - 17
DISASSEMBLER	18 - 22
MN6502	23 - 23
MNZ80A	24 - 25
MN6800	26 - 27
FILE.COPIER	28 - 30
ASSEMBLER	31 - 36
ASSEMBLER.IN	37 - 46
ASSEMBLER.OUT	47 - 49
TEST.DISK	50 - 53
ANSRTRAN.SRC	54 - 57
ANSRTRAN.OBJ	58 - 58
ANSRTRAN.PGM	59 - 65
BUFFER	66 - 66
TEMPORARY.PROG	71 - 76

DATA CONVERSION Between

OSI FLOPPY DISK
(OS-65U)

Write for a quote:

DBMS, Inc.
PO Box 347
Owings Mills, MD 21117

&

IBM FORMAT FLOPPY DISKS

9-TRACK MAGNETIC TAPE

PUNCHED CARDS

OSI-FORTH 2.0 / FIG-FORTH

OSI-FORTH 2.0 is a full implementation of the FORTH Interest Group (FIG) version of the FORTH language. It conforms to the FORTH-79 Standards Specification, and includes a resident text editor and 6502 assembler as part of the basic package. The OSI-FORTH system runs under the OS-65D3 operating system on any disk-based Ohio Scientific system, and has access to all DOS commands and resources. The package price is \$79.95 for either a 8" or mini disk and manual. The manual is available separately for \$9.95, deductible later from package purchase price.

Mr. Clean

Let Mr. Clean pretty up those BASIC programs! This very fast machine language program will resequence (renumber) and/or pack (remove spaces and REMarks), from any OS-65D3 BASIC program. Mini or 8" disk and instructions is \$17.95.

VIDEO BANNER

Video Banner puts up to three lines of nine letters and/or punctuation marks each on the OSI 540 video screen. The letters are '3-D', built from graphics elements (resembling letters above). Mini or 8" disk contains the data files for the characters, utility programs to display them on the screen and save screens to disk, and demo program. Documentation provided will enable you to incorporate the data base and display into application programs (educational programs, games, clock programs, display 'signs'). An excellent program for dealers to catch customer eyes! Mini or 8" disk is \$24.95.

ADVENTURE-65

Explore a deep and mysterious cave, but watch out for the dragon and don't spill the lamp oil! Will you find all of the treasure before the lamp burns out? Will your leap across the chasm be successful? How do you deal with the dragon? Can you find all of the rooms in the cave? This micro version of the famous 370 Adventure retains all of the excitement and frustration of the original. No two trips into the cave will be exactly alike, so keep exploring! Requires 8K of memory. Cassette is \$9.95, mini or 8" disk is \$14.95.

DISK-PAK 1

A collection of very useful OS-65D3 utilities. DSKPRP prepares a new diskette for use, by initializing any track range and putting in a blank directory. DIR540 uses the full 540 video screen (or terminal) to display the complete sorted directory in one view, allowing file creation with the full directory in view. SAVE automatically creates a file of given length by finding available space for you! It will ALSO optionally figure out the space needed to save whatever program is in a scratch file ("SCR"), create the file and transfer the program from SCR to the new file! All of the convenience of an Apple with the simple sequential files of OSI! COPIER copies the OS-65D3 System onto a new disk with a minimum of disk swaps (uses 24K). ONERR is a program (subroutine) that modifies OS-65D3 to allow your BASIC program to automatically execute any given line number upon detection of an error. BACKUP maintains a backup directory on disk and will restore a directory lost due to a disk error or mistake. Mini or 8" disk is \$24.95.

MANUALS

C4P/C4P-MF Users Manual: This is a nicely typeset manual recently released by Ohio Scientific. \$8.95

OSI-SAMS Service Manuals: These excellent manuals contain schematics, logic diagrams, parts lists, scope traces and other valuable service data. C4P/C4P-MF manual is \$14.95 (C4 manual for C8, too). C1P/SB11 is \$7.49 C3/C20EM manual is \$38.95.

CONNECTORS

A set of four male and four female connectors of the type used by OSI for board connection to the backplane and other I/O. \$4.95

INFO

Our Floppy Disk Information pack provides the information you need to upgrade your OSI ROM BASIC system to a disk-based system. \$3.49

** Allow 30 days for delivery. Florida residents add 4% Sales Tax. Specify system model. **

Send for our new FREE catalog containing many programs for ROM or disk systems, information packs, hardware, and other useful goodies for Ohio Scientific systems!

TECHNICAL PRODUCTS COMPANY
P. O. BOX 12983
GAINESVILLE, FLORIDA 32604

LETTERS

ED:

STRING BUG REVISITED

As everyone else with BASIC-in-ROM, I have been bitten by "Screenflickerus stringarray-us" and I have appreciated the articles by Mark Minasi and Stan Murphy on the bug killers. I was, therefore, very interested in E.D. Morris's article on the hardware mod for the garbage collector in your March 1981 issue. But then I read later in the article that he still had problems with the fix so I sat down and ran some of the programs in the article because I haven't had any problems lately.

Lo and behold, the second program ran as listed here:

```
LIST
10 DIM Z$(11)
20 Z$(1)="I LIKE"
30 Z$(2)="MICRO"
40 Z$(3)=Z$(1)+Z$(2)
50 PRINT FRE(8)
60 PRINT Z$(3)
OK
RUN
7256
I LIKEMICRO
OK
```

I do not have a corrected ROM, so that worked as OSI arranged the computer. Since I was checking out the article, now I tried the next program with the proper dimensions included and memory set at 2000.

```
10 DIM A$(5):FOR X=1 TO 255
20 A$(5)=A$(5)+"A"
30 PRINT LEN(A$(5))
40 NEXT
```

This program went past 46 on the print out; in fact, it went to 255 and the OK prompt at completion.

Now I won't believe that Mr. Morris is wrong, but the "Minasi bug exterminator" works to perfection on my computer. I have even used it in a more complex way on a recipe program that I wrote for my wife in which there is a whole lot of string manipulations and it still doesn't hang. I wonder if OSI's supplier made a change between the time he bought his computer and when I bought mine (a C4P purchased Feb. 1980), such that my BASIC calls the garbage collector (flawed though it is) more often than his. Has anybody else found this difference?

The only time I have problems with "Screenflickerus stringarray-us" is when changing strings around and making a string in an array with one number equal to a string in the same array with a larger number. My BASIC seems to open a space for 33 characters in the smaller numbered string of the array. If the resultant string is longer than 33, the system hangs, however; if the resultant string is smaller, you can get some very interesting results as this program does:

```
LIST
5 DIM A$(5)
7 A$(0)="WET"
10 A$(1)="I LIKE"
20 A$(3)="PEEK(65) "+A$(0)
22 REM A$(3) AT THIS STAGE IS "PEEK (65) WET"
25 A$(3)=" TO READ "
```

```
30 A$(2)=A$(1)+A$(3)
40 PRINT FRE(8)
50 PRINT A$(2)
60 PRINT A$(3)
OK
RUN
7178
I LIKE TO READ PEEK (65)
TO READ
OK
```

Note if the spaces are not included in line 20 then the end of A\$(3) will have the sign(\$) to the length of 33.

Next I'd like to offer a hex to decimal subroutine which is more versatile than most I've seen and faster than the one in Ed Carlson's book even for a 4 digit hex number. It is twice as fast for a two digit hex number.

```
LIST
10 INPUT"HEX NUMBER ANY LENGTH"; HEX$
20 GOSUB 9000
30 REST OF PROGRAM
9000 N=0:H=1:FOR X=1 TO LEN(HEX$):D=ASC(RIGHT$(HEX$,X))-48
9010 D=D+7*(D>9):N=N+D*H:H=H*16:NEXT:PRINT N:RETURN
```

DALE MAYERS
LANSING, MO 49810

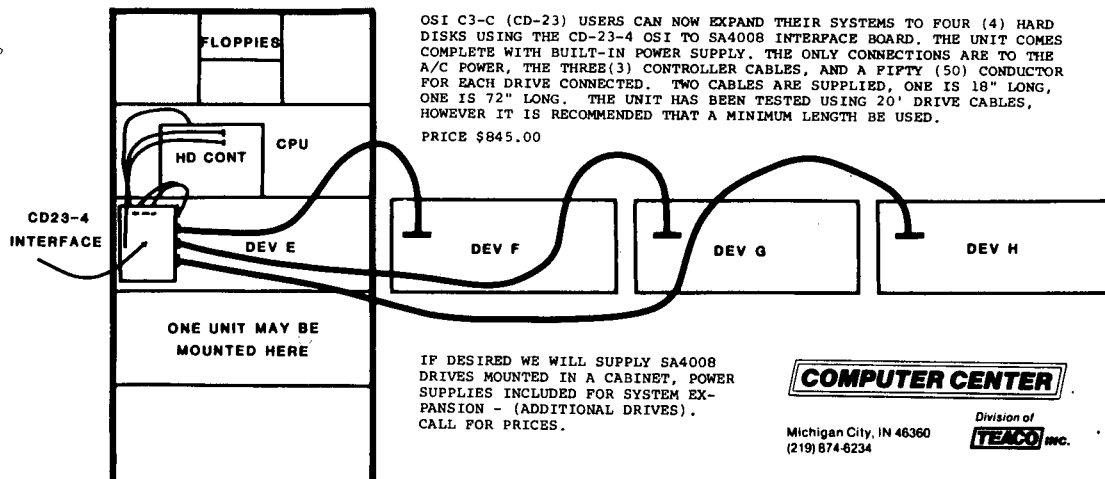
DALE:

My C3D doesn't have any problem with string arrays and doesn't lose track of strings as your machine does. I wonder how many different systems with OSI on the front are really out there...

Your Hex convertor is the neatest I've seen. I don't understand the (D>9) in line 9010, but it works!

AL * * * * *

(4) HARD DISKS ON (1) OSI COMPUTER



ED:

I have a few comments to add to the "Garbage Collector" article which appeared in the March 1981 issue of Peek (65). The ROM routine starts at \$B147 and ends at \$B24C. There is a typo in the original article.

I received a letter from a company selling replacement ROMs suggesting my test programs are not "fair". My tests place an "undue stress" on the garbage collector and are no indication of performance in a "real" program.

My response to this is that these tests are written in valid BASIC code and thus should run. In fact, they do run on OSI disk BASIC and on a PET computer.

The following is another program to demonstrate problems in the original garbage collector and with many of the "fixes".

```
5 DIM A$(11)
10 FOR X= 65 TO 70
20 A$(0) = A$(0) + CHR$(X)
30 PRINT A$(0)
40 NEXT
```

The last line of the printout will be "ABCDEF". Now add a line calling the garbage collector: 35Z =FRE (9) and the program will no longer work correctly. This happens only when zero subscripts are used. The garbage collector does not handle zero subscripts properly. The following code, when put into EPROM to replace the BASIC 3 ROM, will pass all of my garbage collector torture tests. I have not been able to make it fail. This code appears to be a total solution.

7751 - 262

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
B140									85	A5	86	86	81	85	82	A0
B150	00	84	9D	A5	7F	A6	80	A6	AA	86	AB	A9	68	85	71	84
B160	72	C5	65	F0	05	20	D9	B1	F0	F7	A9	06	85	A0	A5	7B
B170	A6	7C	85	71	86	72	E4	7E	D0	04	C5	7D	F0	05	20	D3
B180	B1	F0	F3	85	A4	86	A5	A9	04	85	A0	A5	A4	A6	A5	E4
B190	80	D0	07	C5	7F	D0	03	4C	18	E2	85	71	86	72	A0	01
B1A0	B1	71	08	C8	B1	71	65	A4	85	A4	C8	B1	71	65	A5	85
B1B0	A5	28	10	D7	C8	B1	71	A0	00	0A	69	05	65	71	85	71
B1C0	90	02	E6	72	A6	72	E4	A5	D0	04	C5	A4	F0	C1	20	D9
B1D0	B1	F0	F3	C8	B1	71	10	30	C8	B1	71	F0	2B	C8	B1	71
B1E0	AA	C8	B1	71	C5	82	90	06	D0	1E	E4	81	B0	1A	C5	AB
B1F0	90	16	D0	04	E4	AA	90	10	86	AA	85	AB	A5	71	A6	72
B200	85	9C	86	9D	88	88	84	A2	A5	A0	18	65	71	85	71	90
B210	02	E6	72	A6	72	A0	00	60	C6	A0	A6	9D	F0	F5	A4	A2
B220	18	B1	9C	65	AA	85	A6	A5	AB	69	00	85	A7	A5	81	A6
B230	82	85	A4	86	A5	20	D6	A1	A4	A2	C8	A5	A4	91	9C	AA
B240	E6	A5	A5	A5	C8	91	9C	4C	4B	B1	EA	EA	EA			

ED:

Please note that in Jim Rajac's letter (Vol. 2, #4) that the command,DISK!"IO,01" will not work! There must be a space between "IO" and "01" (DISK!"IO 01") no comma please!!!

If your readers would like to experiment with an underlying cursor then POKE 9682, 64 (POKE 9684,32 for ClPMF). Using the much annotated POKE 9680(9682 for ClPMF) you can change to a more suitable cursor character. I suggest that you try 16, 94 or 172 for an underlying cursor.

The following two short programs illustrate using the command 'DISK!"MEM NNNN,NNNN"' as a print at command. Thank you for a great OSI resource!

```
5 POKE9800,255:PRINT:PRINT
:PRINT:POKE9800,32
10 DISK!"MEM D000,D000"
20 PRINT#5,"*****"
:PRINT#9
25 PRINT#9
26 T=T+1:IFT=50THEN40
30 GOTO20
40 DISK!"MEM D20A,D20A"
50 PRINT#5,"CURTAIN CALL";
:PRINT#9
60 GOTO60
```

```
1 A=20
5 POKE9800,255:PRINT:PRINT
:PRINT:POKE9800,32
6 A$="MEM D1E9,D1E9"
10 DISK!A$
20 PRINT#5,"* * * * *";
25 PRINT#9
26 FORT=1TOA:NEXT
30 DISK!A$
40 PRINT#5,"* * * * *";
45 PRINT#9
46 FORT=1TOA:NEXT
50 GOTO10
```

These programs are for the ClPMF. Screen locations (DNNN) will have to be changed for other systems, also the quick screen clears need to be changed (LINE #5 in each program) to suit other systems.

Charles Lundberg
Olympia, WN 98502

**LOW low PRICED
I/O BOARD**

PLUGS INTO ClP OR S.B.II

THIS BOARD WHICH PLUGS INTO THE EXPANSION SOCKET UTILIZES A 6522 V.I.A. AND 3 DECODING CHIPS. THE 6522 HAS 16 I/O LINES INDIVIDUALLY PROGRAMABLE AS INPUTS OR OUTPUTS PLUS 4 CONTROL OR HANDSHAKING LINES, ONE SERIAL I/O LINE, 2 TIMERS AND INTERRUPT CAPABILITY. THE BARE BOARD IS \$11.95, A KIT IS \$24.95 AN ASSEMBLED AND TESTED BOARD IS \$49.95. BOARDS ARE STACKABLE FOR MORE I/O LINES. THEY ARE DRILLED AND TINNED. FOR MORE INFO SEND S.A.S.E. TO MARTIN MONTGOMERY 905 MIDDLE DR. NEW WHITELAND, IN 46184

AD\$

BIG BAG \$25 (BIGGEST BAG (5 1/4" DISK) of software you can buy for the bucks) 1) Single Disk Copier. 2) Terminal Emulator makes a terminal out of your computer. 3) High Memory Disk Buffers. 4) Data Management System (not OSI). 5) Disk Head Load-Unload for all disk I/O. 6) Spool to Disk (better than indirect memory). 7) VTOC Fixer. 8) Menu of Files at Boot Time. 9) File Transmission Program via phone lines with error checking and recovery. New PERSWP personal word processor for OS65D ClP/4P. PERSWP is \$15 with BIGBAG, \$20 sold separately. For OS65D ClP or C4P. Detailed documentation is available for \$5, refundable when software is purchased. Computer Power 3223 Suffolk Lane Fallston, MD 21047 Phone 301-692-6538

FOR SALE: 48K C8P-DF with OSI Color Monitor. Moving up to C3. Excellent condition. \$2800.00. T.J. Insulation, Inc., 7166 High St., Lima, NY 14485.

Earl D. Morris
Midland, MI

OSI

AARDVARK NOW MEANS BUSINESS!

OSI**WORD PROCESSING THE EASY WAY—
WITH MAXI-PROS**

This is a line-oriented word processor designed for the office that doesn't want to send every new girl out for training in how to type a letter.

It has automatic right and left margin justification and lets you vary the width and margins during printing. It has automatic pagination and automatic page numbering. It will print any text single, double or triple spaced and has text centering commands. It will make any number of multiple copies or chain files together to print an entire disk of data at one time.

MAXI-PROS has both global and line edit capability and the polled keyboard versions contain a corrected keyboard routine that make the OSI keyboard decode as a standard typewriter keyboard.

MAXI-PROS also has sophisticated file capabilities. It can access a file for names and addresses, stop for inputs, and print form letters. It has file merging capabilities so that it can store and combine paragraphs and pages in any order.

Best of all, it is in BASIC (OS65D 51/4" or 8" disk) so that it can be easily adapted to any printer or printing job and so that it can be sold for a measly price.

MAXI-PROS — \$39.95

**THE EDSON PACK
ALL MACHINE CODE GAMES
FOR THE 8K C1P**

INTERCEPTOR — You man a fast interceptor protecting your cities from Hordes of Yucky Invaders. A pair of automatic cannon help out, but the action speeds up with each incoming wave. It's action, action everywhere. Lots of excitement! \$14.95

MONSTER MAZE — An Arcade style action game where you run a maze devouring monsters as you go. If one sees you first, you become lunch meat. Easy enough for the kids to learn, and challenging enough to keep daddy happy. \$12.95

COLLIDE — Fast-paced lane-switching excitement as you pick up points avoiding the jam car. If you succeed, we'll add more cars. The assembler code provides fast graphics and smooth action. \$9.95

**SPECIAL DEAL—THE ENTIRE EDSON PACK—
ALL THREE GAMES FOR \$29.95**

THE AARDVARK JOURNAL

FOR OSI USERS — This is a bi-monthly tutorial journal running only articles about OSI systems. Every issue contains programs customized for OSI, tutorials on how to use and modify the system, and reviews of OSI related products. In the last two years we have run articles like these!

- 1) A tutorial on Machine Code for BASIC programmers.
- 2) Complete listings of two word processors for BASIC IN ROM machines.
- 3) Moving the Directory off track 12.
- 4) Listings for 20 game programs for the OSI.
- 5) How to write high speed BASIC — and lots more —

Vol. 1 (1980) 6 back issues - \$9.00

Vol. 2 (1981) 2 back issues and subscription for 4 additional issues - \$9.00.

ACCOUNTS RECEIVABLE — This program will handle up to 420 open accounts. It will age accounts, print invoices (including payment reminders) and give account totals. It can add automatic interest charges and warnings on late accounts, and can automatically provide and calculate volume discounts.

24K and OS65D required, dual disks recommended. Specify system.

Accounts Receivable. \$99.95

***** SPECIAL DEAL — NO LESS! *****

A complete business package for OSI small systems — (C1, C2, C4 or C8). Includes MAXI-PROS, GENERAL LEDGER, INVENTORY, PAYROLL AND ACCOUNTS RECEIVABLE — ALL THE PROGRAMS THE SMALL BUSINESS MAN NEEDS. \$299.95

P.S. We're so confident of the quality of these programs that the documentation contains the programmer's home phone number!

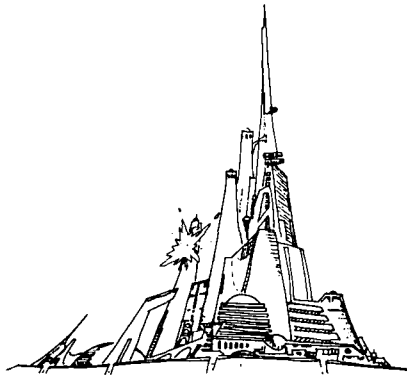
SUPERDISK II

This disk contains a new BEXEC* that boots up with a numbered directory and which allows creation, deletion and renaming of files without calling other programs. It also contains a slight modification to BASIC to allow 14 character file names.

The disk contains a disk manager that contains a disk packer, a hex/dec calculator and several other utilities.

It also has a full screen editor (in machine code on C2P/C4)) that makes corrections a snap. We'll also toss in renumbering and program search programs — and sell the whole thing for — SUPERDISK II \$29.95 (5 1/4") \$34.95 (8")

AND FUN, TOO!

**BOOKKEEPING THE EASY WAY
— WITH BUSINESS I**

Our business package 1 is a set of programs designed for the small businessman who does not have and does not need a full time accountant on his payroll.

This package is built around a **GENERAL LEDGER** program which records all transactions and which provides monthly, quarterly, annual, and year-to-date PROFIT AND LOSS statements. GENERAL LEDGER also provides for cash account balancing, provides a BALANCE SHEET and has modules for DEPRECIATION and LOAN ACCOUNT computation. GENERAL LEDGER (and MODULES) \$129.95.

PAYROLL is designed to interface with the GENERAL LEDGER. It will handle annual records on 30 employees with as many as 6 deductions per employee. PAYROLL - \$49.95.

INVENTORY is also designed to interface with the general ledger. This one will provide instant information on suppliers, initial cost and current value of your inventory. It also keeps track of the order points and date of last shipment. INVENTORY - \$59.95.

GAMES FOR ALL SYSTEMS

GALAXIAN - 4K - One of the fastest and finest arcade games ever written for the OSI, this one features rows of hard-hitting evasive dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. Specify system — A bargain at \$9.95

MINOS - 8K - — Features amazing 3D graphics. You see a maze from the top, the screen blanks, and when it clears, you are in the maze at ground level finding your way through on foot. Realistic enough to cause claustrophobia. — \$12.95

NEW — NEW — NEW

LABYRINTH - 8K - This has a display background similar to MINOS as the action takes place in a realistic maze seen from ground level. This is, however, a real time monster hunt as you track down and shoot mobile monsters on foot. Checking out and testing this one was the most fun I've had in years! — \$13.95.

TIME TREK - 8K - Real Time and Real graphics Trek. See your torpedoes hit and watch your instruments work in real time. No more unrealistic scrolling displays! — \$9.95

SUPPORT ROMS FOR BASIC IN ROM MACHINES — C1S/C2S. This ROM adds line edit functions, software selectable scroll windows, bell support, choice of OSI or standard keyboard routines, two callable screen clears, and software support for 32-64 characters per line video. Has one character command to switch model 2 C1P from 24 to 48 character line. When installed in C2 or C4 (C2S) requires installation of additional chip. C1P requires only a jumper change. — \$39.95
C1E/C2E similar to above but with extended machine code monitor. — \$59.95

Please specify system on all orders

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our \$1.00 catalog lists it all and contains free program listings and programming hints to boot.

**OSI**

AARDVARK TECHNICAL SERVICES, LTD.
2352 S. Commerce, Walled Lake, MI 48088
(313) 669-3110

**OSI**

ED:

For those of us BASIC-in-ROM users who get tired of seeing the famous 'OK' prompt print on the screen after every immediate mode command, I have a routine to change the prompt:
Enter this through monitor:

```

0003 4CF01C JMP $1CF0
      ;OR WHEREVER YOU DESIRE THE
      ROUTINE
.....
.....
1CF0 A200 LDX #000
1CF2 BDE01C LDA $1CE0,X
      ;OR FURTHER BACK IF MESSAGE
      IS LONGER
1CF5 F006 BEQ $1CFD
      ;ZERO FOUND AT END OF
      MESSAGE
1CF7 20EEFF JSR $FFEE
      ;SEND TO OUTPUT WHEREVER IT
      IS
1CFA E8 INX
1CFB D0F5 BNE $1CF2
      ;MORE TO PRINT
1CFD 4CC9A8 JMP $A8C9
      ;JUMP HERE TO BYPASS THE
      'OK' PRINTING

```

A \$00 must follow the final character in the message, like the example:

```

1CE0 0D ;LINE FEED AND
      CARRIAGE RETURN HELP
1CE1 0A
1CE2 53 ;'R
1CE3 45 ;'E
1CE4 41 ;'A
1CE5 44 ;'D
1CE6 59 ;'Y
1CE7 0D ;LF AND CR AT END OF
      LINE
1CE8 0A
1CE9 00 ; ZERO AT END OF
      MESSAGE

```

Now you can have "YOUR NEXT COMMAND, SIRE?", "KILL THE RED DRAGON!!", or whatever as a command prompt.

I suppose that this would work with 65D and 65U if the entry point to avoid the prompt was known (Software Consultants' disassembly manual would come in handy here!).

Would people be interested enough in an 8-bit DAC project for the ClP or Superboard II?

Curtis R. Anderson
Randolph, NY

Curtis:

Sure they would!

AL

ED:

PEEK (65) has provided me with several useful tips, but I do have a few problems to which I have been unable to figure out solutions. First of all, my system is: OS Challenger II with dual floppy-disk drive; Hazeltine 1500 input-output video and NEC Spinwriter printer. Programs are BASIC OS-65U level 1 and WP-2 word processor. My specific problems with WP-2 are: The command DIRECT does not print out on the printer after P is entered in response to the console/printer question. I have tried various POKE's for 17920, which seems to have something to do with the print-out, but none put it on the Spinwriter.

I cannot get my multi-lingual thimble to print the special characters such as accent marks and inverted punctuation. The Spinwriter manual (Appendix II) talks of using a tilde (~) followed by a DCI code, but I have found no combination of ESCAPE's or CONTROL plus tilde or other input that does the job.

My one problem with BASIC is that the printer stops at the end of each page of print-out. What is the POKE to cause the printer to continue on with the next page (after proper spacing)? The manuals say CONTROL-W can be used, but I am either entering it at the wrong time or for some other reason it does not work.

Really hope some reader can provide some useful solutions or suggestions.

O.G. Haywood
Vero Beach, FL

Mr. Haywood:

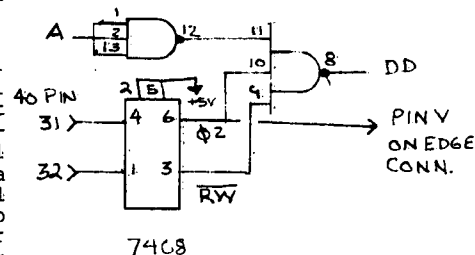
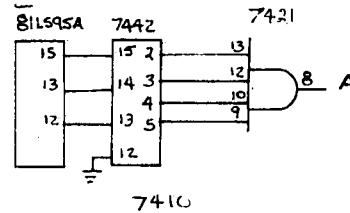
Is your NEC a serial or parallel model? If serial, through the 550 board, there are some changes to WP-2 you can make noted in Tech Newsletter #28 (see your dealer) which might help. We print Peek (65) on a serial NEC with WP-2, so I know it can be done. Have your dealer contact DBMS, INC. if he can't fix you up. Anyway, our WP-2 asks "list on line Printer instead of here?" and we must answer Y, not P, to go to the printer!

Who else has solved this problem, and who knows anything about the printer stopping at page end in BASIC?

AL

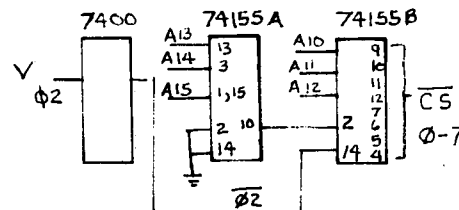
ED:

If any of your readers have tried using the expansion interface in my article in the Peek (65) issue dated August 15th (issue Vol. 1, #8), they may have found it didn't work. That's because the DD line must be a product of the expansion address and 02 and R/W. Below is the schematic I am now using.



Builders should delete the 7402 and the 7445 and the 2K OHM resistor from the article.

On the RAM board, use the following schematic to enable the 2114's.



My apologies for whatever frustrations my errors have caused.

Bruce Showalter
Abilene, TX

ED:

Concerning the OSI standard protocol:

1. For sending programs, two forms should be devised, a) straight ASCII text, b) tokenized as in memory - Aardvark does it that way, both should have some type of checksum/verification.

2. For sending text, straight ASCII alone.

3. I suggest standard speeds of 300/1200 baud as used with the penril modem (not bell 202 at 1200 baud).

4. How about the HDLC "packet" method? See QST magazine (HAM radio Magazine) April 1981, page 27.

5. What are other groups (non-OSI) using? One of OSI's biggest problems has been to be different than everyone else, even if they have a better way. Let's not make the same mistake!

I am planning a new monitor ROM or actually planning how OSI should have done it. (No fancy frills, no fancy anything, just what "should" have been there from the beginning) I am asking for help with this from everyone and would like to "share" the results of our labor with the readers of Peek (65). I am volunteering to be coordinator of this project and will produce prototypes and insure proper documentation. Nothing will happen unless the readers help. Here is what I have so far; Everyone send me your "fixes" and routines, etc. Improvements/fixes to OSI ROM's must be compatible with standard system (C4P).

1. Standard keyboard routine - no more shift lock mess, a) routine should have entryptoint that waits for character as standard OSI routine does, b) entry point that returns with or without a keypress (returns with null if no key pressed).

2. Machine language screen clear or load as subroutine (POKE value to be loaded, 32 clears screen).

3. Color clear or load like #2 above, note: entry point available for #2 that automatically does #3 also.

4. Shift "O" erases character - poke to allow standard method for people that have serial terminals hooked up. Poke to change this character to something else, (like rub out).

5. Shift "P" erases line not just do a <CR>.

6. Input vectors like ClP (or should we start with ClP ROMS and convert them to C4P?).

7. GARBAGE COLLECTION FIX.

8. Error messages WITHOUT graphics.

9. Simple line editor - like the one in 65U.

10. Adjustable null count when using cassette/printer port, not forced to use a minimum of 10.

11. Print #1, print #2 as in OS65U/65D. The regular "print" prints to the screen, print #1 to the cassette port, print #2 put character in accumulator and jumps to a user supplied subroutine for driving other printers, plotters, or anything else. Or a POKE to do same as above.

12. Do same for inputs as in #11.

13. POKE to allow leading spaces on inputs.

14. 9 or 12 digit math extensions pokeable to integer only math for speed or add integer math to BASIC.

We must first rate these as to importance, difficulty, and in what order they should be implemented.

Tom Westhoff
Wilmar, MN

Tom:

We are with you 100%.

AL

* * * * *

ED:

I would like to recommend to my fellow OSIers, a piece of software known as MULTIPLE PARTITION SYSTEM UTILITIES (VERSION SE:1.5 12/12/79), as a diagnostic tool to test for bad chips and what have you. I must admit that I got this in a devious way (someone gave it to me) and I have no idea whether this is OSI or issued by another firm, and it doesn't say.

In late February, I decided to upgrade from a ClPMF to a C3S1, particularly since the OSI dealer I have been working with is a good friend and among all of OSI's problems has been a god-send whenever I got cornered into a situation I could not solve. When I got the C3S1, there were problems almost immediately; we thought it was a disk drive problem but this theory was later discounted because some things worked and others didn't. Then I got the MP Util disk (the program only occupies track 0), and I was able to determine that one particular location consistently indicated a problem, viz. data chip D4 on the 3rd row of the 3rd board (32-48K RAM partition). When several chip replacements, however, did not solve the problem, we decided to replace the whole board and that solved the problem

nicely, plus a good cleaning of the drives and heads.

This was, however, not the end of this debacle... because my software (particularly DMS and 65U which I use most often) had been exhibiting constant disk copy errors, for instance checksum errors at anywhere between 268000 upwards. So back to my dealer to obtain a clean copy of DMS and 65U (bear in mind that the original disks which I got with the system also had checksum problems). This new DMS 9/79 apparently has slight variations from the DMS 9/79 I had before... the new edition has a HOW-TO scratch file in the directory; the previous or original disk didn't, but the menu listing in both state 9/79! Strange!

Throughout this, I was having problems with losing first names after sorts or repacks, where they were entered with a comma, like so: Smith, John A; apparently commas acted as delimiters... I even wrote OSI about a fix, and lo and behold, I got an answer (miracles never cease) in the form of a partial Xerox of a Tech newsletter, indicating a poke which has to be fixed, viz. POKE 2976,58;POKE 2972,44 -Should both be changed to 13! As far as I can determine (I made a printout of all the source listings quite a while ago) the following are the offending line numbers, in the DMS 9/79 nucleus disk: Mailer 51120; Merger 60; Stat03 50170; Stat3a 51030; Insert 51120, IFGETI 51115, and in the 9/79 Sort disk lines 51115 in SORT and KSORT. Anyone have any ideas about other locations I may have missed? It seems strange to me that OSI has some pokes as enabling and in other programs on the same disk, pokes as disabling commas, semicolons etc. Having fixed all that, things seem to work reasonably well. One wonders seriously though, that with all this expensive software, they cannot come up with a more efficient sort program. The OS-DMS SORT 9/79 is slow and tedious because it is a simple single-level bubble sort. Sorting a 4 character-alphanumeric field in a keyfile, took all of 6 hours, the other day for a file containing 838 records... that's mighty slow.

I want to commend you for Peek (65), I've been able to pick up little tidbits here and there which proved useful in my own quest with 65U and the C3 from your columns and letters to the editor. I have

to rewire my RS232 (no time yet) so that I do not have to unplug my Hazeltine 1520 each time I want to use the modem and also figure out how to make the Hazeltine act as a dumb terminal with the modem, then I can make use of your CBBS instead of writing these lengthy epistles. Telephone rates what they are, I'll no doubt be much less long-winded! One more word on the sort, people will think why I haven't looked at other sort utilities... the main problem is that I am working with some good sized address files and with 2 drives, the KEY file must be on the same disk as the MASTER file, because the OTHER drive can only be used for the DMS SORT disk (because of the interaction with its scratch file).

I am eager to see a column on CP/M, I am just getting my act together, i.e. reading through the several notebooks of documentation. Haven't decided yet if it is worthwhile to redo part of my larger data files for CP/M or to leave them in DMS.

Fred S. Schaeffer

Fred:

I haven't seen the MPS Utilities, but they sound great. (Does anyone know more about them?) I wonder who wrote them, how many hours he worked on them and whether he lost money when someone "gave" you a copy, a complex problem.

In Microsoft BASIC, commas and colons are string delimiters. That means you can use DATA statements like DATA Smith, Jones, Brown, and READ or INPUT statements like INPUT A\$, B\$, C\$, This is achieved by a part of the BASIC interpreter which checks each character input to see if it is a carriage return (ASCII 13) comma (44) or colon (58). The two POKES you mention alter this program so it checks for a CR all three times, but then INPUT % 1, FO\$, PW\$ will not work! So be sure to look at your programs carefully, and POKE the 44 and 58 back as needed.

DMS Sort is slow because it is written in BASIC. The machine language sort in KYUTIL would solve all your DMS sort problems instantly.

I am anxious to start writing the CP/M column... but we are all being delayed by software vendors who are very slow to release review copies of their wares; and Peek (65) is not

rich enough to buy "one of each" to write about them!

AL

ED:

The enclosed program was developed for a hard copy duplication of the ASCII data on the video display of a C1. The original application was to provide a copy of data entered after prompts and instructions. In this instance a P.O.S. terminal built around the C1. I have since used the printouts in a guide for operation, troubleshooting, and repair of the system. It has been used for duplication of results, some good, some bad, of video displays that could not be duplicated using the 'SAVE' command.

Line 10000 - sets parameters
 Q= ADDR 61440 is the C1 status reg of the ACIA, tape & print port
 Q+1= 61441 ACIA data register
 L= Video RAM ADDR for C1 to be PEEKed
 H= Horz position by G

Line 10100 - using the 'WAIT' command to check the status - output 1st char of video RAM increment H (Horz position within a line) go to 10300 if at the end of a line.

Line 10200 - next horz position, if at the end of the video RAM GOTO 10400

Line 10300 - at the end of each video line the printer requires a carriage return and line feed. If R (data register of ACIA) is POKED with 13 a CR is printed; 10 a LF. Horz position H, is reset to zero.

Line 10400 - Return - from sub to next line after call

By changing ACIA ADDR, Video RAM ADDR, and line length, a C4 or C2 may use this routine.

As you can see I'm in dire need of a Word Processing system. Since this routine is embedded within a number of programs, the line may differ from the copy first submitted.

C1 SCREEN PRINT SUBROUTINE

This subroutine is used to provide an exact record of the ASCII data on the video display. The video RAM is PEEKed and applied to the ACIA one location at a time. At

the end of the line a CR & LF are printed. Sample printout shows the results of the subroutine when called.

```
C1 "PRINT SCREEN"
(1) SEARCH (2) RENUMBER
(3) TABLE? 3
LINE 10000
LINE 10100
LINE 10200
LINE 10300
LINE 10400
[LINE 60000 -UTILITY]
  Q 10000 10100 10300 Q=ACIA
  ADDR FOR C1[STATUS]
  R 10000 10100 10300 R=ACIA
  ADDR FOR C1[DATA]
  H 10000 10100 10300 H=HORZ
  BIT POS OF VIDEO LINE
  L 10000 10200 L=VIDE
  O RAM ADDR
  G 10000 10100 G DATA
  VALUE OF VID RAM ADDR
```

```
PRINTOUT USING "PRINT SCREEN"
C1 "SCREEN PRINT" SUBROUTINE
10000 Q=61440:R=Q+1:H=0:FORL=5
3246TO4272:G=PEEK(L)
10100 WAITQ,2:POKER,G:H=H+1:IF
H>32THENGOTO10300
10200 NEXT:IFL>54271THENGOTO10
400:GOTO10100
10300 WAITQ,2:POKER,13:WAITQ,2
:POKER,10:H=0:GOTO10100
10400 RETURN
```

```
REGULAR $? 1.259
UNLEAD $? 1.299
GASAHOL $? 1.315
ENTER R FOR REGULAR
U FOR UNLEADED
G FOR GASAHOL
PUMP#1? R
PUMP#2? U
PUMP#3? G
PUMP#4? R
PUMP#5? R
PUMP#6? G
PUMP#7? U
PUMP#8? R
IS THE ABOVE OK? YES
```

```
7777777777CLARKS7777777777
- PUMP 1 PUMP 2
$ $
gal gal
PUMP 3 PUMP 4
$ $
gal gal
PUMP 5 PUMP 6
$ $
gal gal
PUMP 7 PUMP 8
$ $
gal gal
```

A. F. White
 Broomfield, CO

```

ED:
10 0000 ;
20 0000 ; OUTPUT PROCESSOR
30 0000 ;
40 0000 ; COPYRIGHT (C) 1981 BY MICHAEL B. CARROLL
50 0000 ;
60 0000 ; ALL RIGHTS RESERVED INCLUDING THAT OF
70 0000 ; DISTRIBUTION FOR PROFIT
80 0000 ;
90 0000 ; PAGE ZERO AREAS
100 0000 ;
102 0000 CURPOS = $F0 CURRENT LINE ADDRESS
110 0000 WORK = $FC WORK AREA
120 0000 ;
130 0000 ; PAGE TWO AREAS
140 0000 ;
150 0000 ;
160 0000 OLDCOL = $0200 CURRENT COLUMN OFFSET
170 0000 OLDCHR = $0201 CHAR OVERLAID BY CURSOR
180 0000 SAVE1 = $0202 TEMPORARY SAVE AREA
190 0000 TEMP = $0207 SUBROUTINE AREA
200 0000 OVECT = $021A OUTPUT VECTOR
210 0000 ;
220 0000 ; CONSTANTS
230 0000 ;
240 0000 LSIZE = 64 TOTAL COLUMNS PER LINE
250 0000 PAGES = 8 # OF 256 BLKS ON SCREEN
260 0000 REST = 10 LEFTMOST VISIBLE COLUMN
270 0000 MAXCOL = REST+46+63 RIGHTMOST VISIBLE COL
280 0000 SPAGE = $D0 TOP PAGE ON SCREEN
290 0000 TROW = $C0 00 OFFSET FOR TOP ROW
300 0000 LPAGE = $D7 00 BOTTOM PAGE ON SCREEN
310 0000 BROW = $40 C0 OFFSET FOR BOTTOM ROW
320 0000 TOP = $D000 SCREEN TOP ADDRESS
0000 B1 = DSPLAY/256 HIGH ADDRESS BYTE
0000 B2 = B1*256 INTERMEDIATE CALC
0000 B3 = DSPLAY-B2 LOW ADDRESS BYTE
330 0000 ;
340 0000 ; EXTERNAL ROUTINES
350 0000 ;
0000 VECTOR = $0000 RESTART VECTOR
360 0000 TAPOUT = $FF6C TAPE OUTPUT ROUTINE
370 0000 ;
380 0000 ; MAIN ENTRY POINT
390 0000 ;
400 0000 ; HERE WE SAVE THE REGISTERS AND DETERMINE WHAT
410 0000 ; ACTION IS TO BE TAKEN BASED ON THE INCOMING
420 0000 ; CHARACTER
430 0000 ;
440 3C00 * = $3C00
450 3C00 ;
460 3C00 8D0202 DSPLAY STA SAVE1 SAVE REGS
470 3C03 48 PHA
480 3C04 8A TXA
490 3C05 48 PHA
500 3C06 98 TYA
510 3C07 48 PHA
520 3C08 AC0002 LDA OLDCOL RESTORE OVERLAID CHAR
530 3C0B AD0102 LDA OLDCHR
540 3C0E 91F0 STA (CURPOS),Y
550 3C10 AD0202 LDA SAVE1 RELOAD CHAR TO SHOW
560 3C13 F068 BEQ SETCOL IT'S A NULL, IGNORE IT
570 3C15 ;
580 3C15 C90A CMP #$0A LINEFEED?
590 3C17 F025 BEQ DOLF YES
600 3C19 C90D CMP #$0D CARRIAGE RETURN?
610 3C1B F05E BEQ DOCR YES
620 3C1D C908 CMP #$08 BACKSPACE?
630 3C1F F070 BEQ DOBS YES
640 3C21 C911 CMP #$11 HOME?
650 3C23 F04E BEQ DOHOME YES
660 3C25 C912 CMP #$12 CLEAR?
670 3C27 F03D BEQ DOCLR YES
680 3C29 C913 CMP #$13 ERASE TO LINE END?
690 3C2B F01A BEQ DOEL YES
700 3C2D C914 CMP #$14 ERASE TO PAGE END?
710 3C2F F021 BEQ DOEP YES
720 3C31 ;
730 3C31 ; THE CHARACTER IS TO BE TREATED AS A NORMAL
740 3C31 ; GRAPHIC. PUT IT ON THE SCREEN AND PERFORM
750 3C31 ; A CARRIAGE RETURN/LINE FEED IF WE ARE AT THE
760 3C31 ; END OF THE LINE
770 3C31 ;

```

Having moved to New Jersey from Oklahoma in October, the value of good packing for computer devices was made painfully clear to me - my tape recorder was completely demolished (I have the shoe box of pieces to prove it) and my printer is still out of commission. If you are moving, pack everything with a lot of padding!!

Here's an idea I'll throw out for anyone who is interested: on many large system BASICS entering a null line in response to an INPUT causes execution to continue with the next statement instead of terminating the program. Any variable(s) specified on the statement retain their previous values. This can be very useful in writing 'user-friendly' programs. It should be possible, using OSI BASIC-in-ROM and/or The First Book of OSI, to front-end the input vector (\$0218-9) and, whenever a -CR- is detected, determine if any text was entered. If no text was input, go to that section of the interpreter responsible for processing 'next statement'. As a help, when the input vector is called from BASIC, register X has the number of characters entered so far. Good luck!

The James Loos article on modifying the C1P display (November Peek (65)) was excellent. Other than a few crossed wires, I had no problems at all and the modification was complete in less than half a day. I used 5.8K resistors instead of the 5.1K called for and pulled the 2114 ram chips from the 610 board and everything still ran correctly.

Unhappily, I have no way of burning new PROMs and so copied the existing output routine to ram and modified it to think it was a C2P. Then I thought, "As long as I have to load a routine into ram, why not write a new output processor to do the things I think it ought to do?" The enclosed listing (which I hope is not too long to publish) is the resulting program. Design goals were:

1. recognize and process existing actions correctly. These are NUL, LF, and CR.
2. recognize and process 5 new actions (detailed below). These are BS, DC1, DC2, DC3, and DC4.

```

3. support the use of a 780 3C31 91F0          STA (CURPOS),Y          PUT IT ON SCREEN
yet-to-be-written PRINT AT 790 3C33 C8          INY                     BUMP TO NEXT COLUMN
statement by allowing text and 800 3C34 C039         CPY #MAXCOL+1          PAST RIGHT EDGE?
the cursor to be placed      810 3C36 D045         BNE SETCOL             NO, ALL DONE, EXIT
anywhere on the screen.      820 3C38 20B43C        JSR LF0                 YES, DO A LINEFEED
                                830 3C3B 4C7B3C        JMP DOCR                AND A CARRIAGE RETURN
                                840 3C3E                ;
Character codes and their    850 3C3E                ; LINEFEED
associated actions are:      860 3C3E                ;
                                870 3C3E                ; CALL THE LINEFEED ROUTINE AND RELOAD THE CURSOR
                                880 3C3E                ; OFFSET
X'00' NUL NULL; ignored, no 890 3C3E                ;
action.                      900 3C3E 20B43C        DOLF JSR LF0           DO A LINEFEED
X'08' BS BACKSPACE; the     910 3C41                ;
cursor is moved one position 920 3C41 AC0002        DOLF2 LDY OLDCOL       RELOAD OFFSET
to the left. If it moves    930 3C44 4C7D3C        JMP SETCOL             AND EXIT
beyond the left margin it
wraps to the right margin of
the previous line. If this
would move it above the top
line, no action is taken - the
cursor remains in its original
location.
X'0A' LF LINE FEED; the     990 3C47                ;
cursor is moved down one line.1000 3C47 A920        DOEL LDA #$20          GET A BLANK
If this would cause it to go1010 3C49                ;
below the bottom line, the   1020 3C49 91F0        ELI STA (CURPOS),Y      BLANK THIS CHAR
screen is scrolled up one    1030 3C4B C8          INY                     BUMP TO NEXT CHAR
line, the bottom line is    1040 3C4C C040        CPY #LSIZE             PAST END OF LINE?
blanked, and the cursor stays1050 3C4E 90F9        BCC ELI                NO, CONTINUE
at its original location.    1060 3C50 B0EF        BCS DOLF2             YES, GO RELOAD OFFSET
X'OD' CR CARRIAGE RETURN; the1070 3C52                ;
cursor is moved to the left 1080 3C52                ; ERASE TO END OF PAGE
margin.                      1090 3C52                ;
X'11' DC1 DEVICE CONTROL 1 1100 3C52                ; THIS IS A SPECIALIZED FORM OF THE 'CLEAR'
(HOME); the cursor moved to 1110 3C52                ; FUNCTION: IT ERASES TO THE END OF THE SCREEN
the top left corner of the  1120 3C52                ; FROM THE CURSOR LOCATION INSTEAD OF FROM THE
screen.                      1130 3C52                ; TOP OF THE PAGE
X'12' DC2 (CLEAR); the     1140 3C52                ;
screen is erased and the     1150 3C52 38         DOEP SEC                CALC # PAGES TO ERASE
cursor is moved to the home  1160 3C53                ;
position.                    1170 3C55 E5F1        LDA #LPAGE+1           LDA #LPAGE+1
                             1180 3C57 AA         SBC CURPOS+1          SBC CURPOS+1
X'13' DC3 (ERASE LINE); the 1190 3C58 A5F0        LDA CURPOS             TAX
line is erased from the cursor1200 3C5A 85FC        STA WORK               SET ARG REG
location to the end of the  1210 3C5C A5F1        LDA CURPOS+1          SET 1ST LINE TO CLEAR
line. No cursor movement    1220 3C5E 85FD        STA WORK+1
occurs.                      1230 3C60 20043D        JSR CLEAR             GO ERASE LINES
X'14' DC4 (ERASE PAGE); the 1240 3C63 38         SEC                    GO RESTORE OFFSET
screen is erased from the
cursor location to the end of
the screen. No cursor
movement occurs.
                                1250 3C64 B0DB        BCS DOLF2
                                1260 3C66                ;
                                1270 3C66                ; CLEAR
                                1280 3C66                ;
                                1290 3C66                ; ERASE THE PAGE AND PLACE THE CURSOR AT THE
The names in parenthesis are 1300 3C66                ; 'HOME' POSITION
my names; the correct ASCII  1310 3C66                ;
names are given in capitals. 1320 3C66 A208        DOCLR LDX #PAGES       SET ARG REG
Without comments the program 1330 3C68 A9D0        LDA #SPAGE            POINT TO TOP OF PAGE
will assemble in an 8K      1340 3C6A                ;
machine. For a virgin ClP,   1350 3C6C A000        LDY #$00
change LSIZE to 32, PAGES to 1360 3C6E 84FC        STY WORK
4, REST to 5, MAXCOL to REST 1370 3C70 20043D        JSR CLEAR
plus 23, TROW to $60, LPAGE to 1380 3C73                ;
$D3, and BROW to $60 (I     1390 3C73                ; HOME
think). For a C2P with a high 1400 3C73                ;
quality monitor, REST is 0,  1410 3C73                ; RESET THE CURRENT LINE TO THE TOP OF THE PAGE
MAXCOL is REST plus 63, TROW 1420 3C73                ; AND DO A CARRIAGE RETURN
is $00, and BROW is $C0.    1430 3C73                ;
                                1440 3C73 A9D0        DOHOME LDA #SPAGE      SET CURRENT LINE TO
                                1450 3C73 85F1        STA CURPOS+1          TOP OF PAGE
The major problem with this  1460 3C77 A9C0        LDA #TROW
processor is that it uses some 1470 3C79 85F0        STA CURPOS
zero page locations and can be 1480 3C7B                ;
safely used only with those  1490 3C7B                ; CARRIAGE RETURN
products that don't require  1500 3C7B                ;
$F0-1 and $FC-D (neither OSI's 1510 3C7B                ; PLACE THE CURSOR AT THE LEFTMOST VISIBLE
extended monitor or assembler 1520 3C7B                ; COLUMN
fit that category). For those 1530 3C7B                ;
programs I still use the     1540 3C7B A00A        DOCR LDY #REST        RESET TO LEFT SIDE
reworked OSI output routine. 1550 3C7D                ;
                                1560 3C7D                ; AT THIS POINT, WE'VE PERFORMED THE REQUESTED
To set up the processor, cold 1570 3C7D                ; ACTIONS AND MUST NOW SAVE THE (POSSIBLY)
start BASIC (allowing enough  1580 3C7D                ; UPDATED CURSOR OFFSET, SAVE THE CHARACTER THAT
room for the processor), load

```

the processor into memory somehow (I use a monitor autoloader tape), and, using the monitor G command, jump to the label COLD (\$3D11 in my assembly). This causes the warmstart vector at \$0000 to be pointed to our routine which will alter the output vector (\$021A-B) to point to the processor and will clear the screen before passing control to the BASIC code. Thereafter, should it be necessary to warmstart, the output vector patch will be done automatically.

From BASIC, you use PRINT statements with CHR\$(xx) to perform the desired function (xx is the decimal value of the desired action).

I've also included a listing of a program that will create a monitor autoloader tape of the output processor once it is in memory.

One last final note (my fingers are tiring) - I was glad to see Peek (65) go to an 8 by 11 format but how are we going to file our old, small copies?

"Hobbitt"
Plainsboro, NJ

OHIO SCIENTIFIC

S-FORTH — a full implementation of Fig-FORTH including editor, virtual disk sub-system, and compatibility with OS65D-3 on 5 1/4" or 8" disk. \$34.95.

Source listing \$24.95.
Both for \$49.95.

TOUCH TYPING MADE EASY — 15 lesson set teaches you to "touch type". Now also available for the C1P. 8K. \$19.95.

TITANIC QUEST — a real time search where you risk your remaining supplies to find the Titanic. 8K. \$6.95.

TEXT EDITOR — the best screen text editor available for OSI C4P, C8P disk systems. \$19.95.

Send for our FREE 14 page software and hardware catalog. Includes photos and complete descriptions of all game, utility, and business software.

Aurora Software Associates
P.O. Box 99553
Cleveland, Ohio 44199
(216) 221-6981

```

1590 3C7D ; WILL BE OVERLAID BY THE CURSOR, AND PUT THE
1600 3C7D ; CURSOR ON THE SCREEN BEFORE WE EXIT
1610 3C7D ;
1620 3C7D 8C0002 SETCOL STY OLDCOL SAVE CURSOR OFFSET
1630 3C80 B1F0 LDA (CURPOS),Y SAVE CHAR THAT WILL
1640 3C82 8D0102 STA OLDCHR BE OVERLAID
1650 3C85 A95F LDA $5F PUT A CURSOR ON
1660 3C87 91F0 STA (CURPOS),Y THE SCREEN
1670 3C89 ;
1680 3C89 68 PLA RESTORE REGS
1690 3C8A A8 TAY
1700 3C8B 68 PLA
1710 3C8C AA TAX
1720 3C8D 68 PLA
1730 3C8E 4C6CFF JMP TAPOUT EXIT TO TAPE ROUTINE
1740 3C91 ;
1750 3C91 ; BACKSPACE
1760 3C91 ;
1770 3C91 ; IF WE ARE NOT AT THE LEFTMOST VISIBLE COLUMN,
1780 3C91 ; BACK UP A CHARACTER. OTHERWISE, IF WE ARE
1790 3C91 ; NOT AT THE TOP OF THE SCREEN, BACK UP A LINE
1800 3C91 ; AND SET THE CURSOR AT THE RIGHTMOST VISIBLE
1810 3C91 ; COLUMN. IF WE ARE AT THE TOP OF THE SCREEN,
1820 3C91 ; IGNORE THE REQUEST.
1830 3C91 ;
1840 3C91 A920 DOBS LDA #$20 BLANK CHAR ON SCREEN
1850 3C93 91F0 STA (CURPOS),Y
1860 3C95 C00A CPY #REST AT LEFTMOST?
1870 3C97 F004 BEQ BS1 YES, BACK UP A LINE
1880 3C99 88 DEY NO, BACK UP A CHAR
1890 3C9A 4C7D3C JMP SETCOL AND EXIT
1900 3C9D ;
1910 3C9D A5F1 BS1 LDA CURPOS+1 AT TOP OF PAGE?
1920 3C9F D004 BNE BS2 NO, CONTINUE
1930 3CA1 A5F0 LDA CURPOS AT TOP OF LINE?
1940 3CA3 F0D8 BEQ SETCOL YES, IGNORE BS
1950 3CA5 ;
1960 3CA5 A038 BS2 LDY #MAXCOL SET AT RIGHTMOST
1970 3CA7 38 SEC BACK UP A LINE
1980 3CA8 A5F0 LDA CURPOS
1990 3CAA E940 SBC #LSIZE
2000 3CAC 85F0 STA CURPOS
2010 3CAE B0CD BCS SETCOL EXIT
2020 3CB0 C6F1 DEC CURPOS+1
2030 3CB2 90C9 BCC SETCOL EXIT
2040 3CB4 ;
2050 3CB4 ; LINEFEED SUBROUTINE
2060 3CB4 ;
2070 3CB4 ; IF WE ARE NOT AT THE BOTTOM OF THE SCREEN,
2080 3CB4 ; MOVE THE CURSOR DOWN ONE LINE. OTHERWISE,
2090 3CB4 ; SCROLL THE SCREEN UP ONE LINE.
2100 3CB4 ;
2110 3CB4 A5F1 LFO LDA CURPOS+1 AT BOTTOM PAGE?
2120 3CB6 C9D7 CMP #LPAGE
2130 3CB8 D006 BNE LF1 NO, GO ON
2140 3CBA A5F0 LDA CURPOS AT BOTTOM LINE?
2150 3CBC C940 CMP #BROW
2160 3CBE F00C BEQ LF2 YES, GO SCROLL
2170 3CC0 ;
2180 3CC0 18 LF1 CLC GO FORWARD A LINE
2190 3CC1 A5F0 LDA CURPOS
2200 3CC3 6940 ADC #LSIZE
2210 3CC5 85F0 STA CURPOS
2220 3CC7 9002 BCC LFX EXIT
2230 3CC9 E6F1 INC CURPOS+1
2240 3CCB 60 LFX RTS ALL DONE, EXIT
2250 3CCC ;
2260 3CCC A207 LF2 LDX #SKELX-SKEL COPY PROTOTYPE
2270 3CCE BDFC3C LF2A LDA SKEL,X INSTRUCTIONS TO
2280 3CD1 9D0702 STA TEMP,X PAGE 2
2290 3CD4 CA DEX
2300 3CD5 10F7 BPL LF2A
2310 3CD7 ;
2320 3CD7 A207 LDX #PAGES-1 OUTER LOOP COUNTER
2330 3CD9 A000 LDY #$00 INNER LOOP INDEX
2340 3CDB 200702 LF3 JSR TEMP GO SCROLL A CHAR
2350 3CDE D0FB BNE LF3 PAGE NOT DONE, LOOP
2360 3CE0 EE0902 INC TEMP+2 POINT TO NEXT PAGE
2370 3CE3 EE0C02 INC TEMP+5
2380 3CE6 CA DEX SCROLL NEXT PAGE
2390 3CE7 D0F2 BNE LF3

```

```

2400 3CE9 ;
2410 3CE9 C040 LF4 CPY #BROW FINISHED BOTTOM LINE?
2420 3CEB F005 BEQ LF4A YES, EXIT
2430 3CED 200702 JSR TEMP NO, SCROLL A CHAR
2440 3CF0 D0F7 BNE LF4 AND AGAIN
2450 3CF2 ;
2460 3CF2 A920 LF4A LDA #$20 GET A BLANK
2470 3CF4 A03F LDY #LSIZE-1 SET LOOP INDEX
2480 3CF6 ;
2490 3CF6 91F0 LF5 STA (CURPOS),Y BLANK BOTTOM LINE
2500 3CF8 88 DEY
2510 3CF9 10FB BPL LF5
2520 3CFB 60 RTS ALL DONE, EXIT
2530 3CFC ;
2540 3CFC ; PROTOTYPE INSTRUCTIONS FOR SCROLLING ROUTINE
2550 3CFC ;
2560 3CFC B940D0 SKEL LDA TOP+LSIZE,Y
2570 3CFF 9900D0 STA TOP,Y
2580 3D02 C8 INY
2590 3D03 60 SKELX RTS
2600 3D04 ;
2610 3D04 ; CLEAR SUBROUTINE
2620 3D04 ;
2630 3D04 ; REG X HAS # PAGES TO BE ERASED
2640 3D04 ; REG Y HAS OFFSET OF FIRST COLUMN TO BE ERASED
2650 3D04 ; WORK HAS FIRST LINE TO BE ERASED
2660 3D04 ;
2670 3D04 A920 CLEAR LDA #$20 GET A BLANK
2680 3D06 91FC C1 STA (WORK),Y BLANK A CHAR
2690 3D08 C8 INY NEXT CHAR
2700 3D09 D0FB BNE C1
2710 3D0B E6FD INC WORK+1 NEXT PAGE
2720 3D0D CA DEX
2730 3D0E D0F6 BNE C1
2740 3D10 60 RTS ALL DONE, EXIT
2750 3D11 ;
2760 3D11 ; THE FOLLOWING CODE IS NOT STRICTLY A PART OF
2770 3D11 ; THE PRINT PROCESSOR, BUT IS THE HOOK INTO
2780 3D11 ; BASIC
2790 3D11 ;
2800 3D11 A501 COLD LDA VECTOR+1 COPY WARMSTART
2810 3D13 8D353D STA JMPOUT+1 ADDRESS TO OUR
2820 3D16 A502 LDA VECTOR+2 VECTOR
2830 3D18 8D363D STA JMPOUT+2
2840 3D1B A202 LDX #$02 COPY NEW WARMSTART
2850 3D1D BD373D PTCH LDA JMPIN,X VECTOR
2860 3D20 9500 STA VECTOR,X
2870 3D22 CA DEX
2880 3D23 10F8 BPL PTCH
2890 3D25 ;
2900 3D25 A93C WARM LDA #B1 POINT OUTPUT VECTOR
2910 3D27 8D1B02 STA OVECT+1 TO US
2920 3D2A A900 LDA #B3
2930 3D2C 8D1A02 STA OVECT
2940 3D2F A912 LDA #$12 CLEAR THE SCREEN
2950 3D31 20003C JSR DISPLAY
2960 3D34 ;
2970 3D34 4CFFFF JMPOUT JMP $FFFF EXIT TO OLD WARMSTART
2980 3D37 ;
2990 3D37 4C253D JMPIN JMP WARM NEW WARM VECTOR

```

LIST

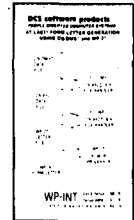
```

10 REM
20 REM GENERATE AUTOLOAD TAPE FOR OUTPUT
30 REM PROCESSOR
40 REM
50 REM REQUIRES THE PROCESSOR TO ALREADY BE
60 REM RESIDENT IN MEMORY
70 REM
80 PRINT CHR$(18)
100 A$="0123456789ABCDEF"
110 B$=CHR$(13)
120 PRINT "START RECORDER AND PRESS <ESC>"
130 IF PEEK(57100)<>222 THEN 130
132 PRINT CHR$(18);
140 C$=".3C00/"
150 GOSUB 500
160 S=3*4096+12*256
170 E=3*4096+13*256+3*16+9

```

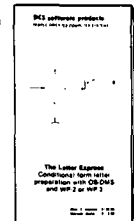
OHIO SCIENTIFIC USERS

FORM LETTER GENERATION WITH
OS-DMS*, WP-2*, and WP-3*



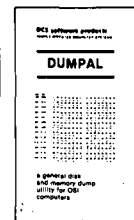
WP-INT V1.2

A form letter utility uniting OS-DMS* and WP-2*. Generates form letters from records stored in OS-DMS data files with the WP-2 word processor. Over 100 satisfied customers world wide. Manual \$2.00. Disk & manual \$79.00.



The Letter Express V1.1

All of the features of WP-INT, plus conditional selection of records from the data-base. Built-in CRT drivers for easy entry and editing. Supports WP-2* and WP-3*. Manual \$2.00. Disk & manual \$129.00.



DUMPAL V1.0

A sophisticated disk and memory dump utility for OS-65D and OS-65U. Prepares reports in ascii and hexadecimal. Reports to console or printer. Supports OS-65D* and OS-65U*. \$30.00.

ASK YOUR DEALER
FOR A DEMONSTRATION.

TO ORDER WP-INT DIRECT
CHECK * C.O.D.
Master Charge * Visa

DCS SOFTWARE PRODUCTS
2729 Lowery Ct.
Zion, IL 60099
(312) 746-8736

FREE OFFER! Send us your name, address, computer configuration, and a list of the software you have purchased for your computer and we will send you DUMPAL free!

DCS software products

PEOPLE ORIENTED COMPUTER SYSTEMS

*OS-DMS, WP-2, & WP-3 sold separately.
*OS-DMS, WP-2, WP-3, OS-65U, & OS-65D
ARE ALL PRODUCTS OF OHIO SCIENTIFIC

WAITING ALL WINTER TO GET YOUR DISK DRIVES BACK?

We offer one week turnaround**

FESSENDENS OFFERS

Complete Service on Current Remex and
Shugart Floppy Disk Drives.

FLAT RATES

8" Double Sided Drive	\$170.00*
8" Single Sided Drive	\$150.00*
5¼" O.S.I. M.P.I. Drive	\$100.00*

*Broken, Bent, or Damaged Parts Extra.

**One Week Turnaround Time Typical

- Fully Equipped Service Facility
- Complete Stock of Parts
- 12 Years Experience in Electronics
- Highly Qualified Technical Staff



Our General Policies

You'll Be Notified of

1. The date we received your drive.
2. Any delays and approximate time of completion.
3. Date Drive was shipped from our plant.
4. Repairs performed on your Drive.
5. Parts used (# and description)
6. Any helpful hints for more reliable performance.

SHIP YOUR DRIVE TODAY

(Via UPS — UPS Blue - Federal Express)
(Include Description of Problem)

TO:

FESSENDEN COMPUTER SERVICE

116 N. 3RD STREET
OZARK, MO 65721
PHONE (417) 485-2501

```

172 T=1
180 FOR I=S TO E
190 ::A=PEEK(I)
200 ::B=INT(A/16)
210 ::C=A-B*16
220 ::C$=MID$(A$,B+1,1)+MID$(
      (A$,C+1,1)+B$

230 ::GOSUB 500.
232 ::T=T+1
234 ::IF T<25 THEN 240
236 :: PRINT CHR$(17) CHR$(19);
238 ::T=0 : TA=TA+3
240 NEXT I
250 C$=".3D1 1G"
260 GOSUB 500
262 PRINT CHR$(18)
270 PRINT:PRINT:PRINT
280 PRINT "TAPE COMPLETED"
290 END
500 PRINT TAB(TA);
502 FOR J=1 TO LEN(C$)
510 ::Q$=MID$(C$,J,1)
520 ::PRINT.Q$;
530 ::IF Q$=B$ THEN PRINT
540 ::WAIT 61440,2
550 ::POKE 61441,ASC(Q$)
560 NEXT J
570 RETURN
OK

```



ED:

In response to Mr. White's request to hear from OSI users (Peek (65) Vol. 2, No. 2, Feb. 81), here are some of my comments and suggestions.

Interstate Computing and Software Corporation is a programming and software development company located in Jenkintown/Abington, just outside of Philadelphia, Pennsylvania. I am Vice President of that company. We have owned and used OSI equipment since March of 1977. Our business is writing software, and consulting, on a number of different computers: WANG 2200 and O.S.I., DIGITAL PDP 11 series, MICRODATA, HONEYWELL and EVOLUTION (Pick Operating System), BURROUGHS B80, B90, & B92, MITS ALTAIR, IMSAI 8080 and DIGI-LOG'S MICROTERM II. Most programs are done in BASIC, although for the BURROUGHS we use COBOL.

There is a strong need for solid, reliable hardware. When OSI was competing with ALTAIR and the old S-100 Bus, the clear winner for reliability was the OSI challenger. Today I see microcomputers put through ordeals that our CHALLENGER, or C-3-C could never survive.

In the four years we have had OSI equipment, we have experienced a number of equipment failures. In all

but a few cases, we have solved them in-house. This kind of down-time is very hard for a client to accept, particularly for someone not familiar with computers (and this is the more probable client for microcomputer).

I read about information found in technical news letters to solve various problems. Occasionally we get them from the dealer. With an aggressive maintenance program, sending hardware-oriented news letters to the maintenance people might be enough (as is done at WANG and DIGITAL), but why doesn't OSI send hardware and software information directly to the user. (DIGITAL sends software updates to paid subscribers, WANG sends to active software vendors and users.) I'm sure most OSI users would be glad to pay for this, (we would!).

There is also a real need for solid software: operating systems, utilities and applications.

After all this, why does ICS continue to use OSI equipment? The first reason is probably familiarity. We know where the problems and weaknesses are, so we (being professional programmers) just program around them. Secondly, OSI still shows the best potential for providing an inexpensive, high volume turnkey system for small businesses. Other microcomputers are entering the Winchester and multiple user field, but are experiencing many of the same problems. This is always to be expected when dealing with state-of-the-art hardware and/or software.

On the hardware side: I would like to see OSI decentralize the CPU requirements (much like WANG has done on their VS computer). Put the 6502 on the disk interface and the I/O ports for more independent control. Add some ultra high speed memory to zero page and the stack. Add wait states to the rest of memory and crank the main CPU board (6502 or whatever) clock up to 4 or 8 MHz. I would like to see more capabilities in the BASIC language for disk interaction. The increased speed would make the generalized utilities (like the report writer) more practical. Large volume backup is definitely needed. The 9 track tape drive is too expensive. Higher density floppies, digital cassette drives, etc., are probably the best solution for today.

Include software for high speed backup!

On the software side: It is imperative to have an ISAM file capability, preferably built in the DOS. It would be helpful to be able to save and retrieve full 8 bit ASCII characters from the disk operating system without resorting to USR(X) DISK I/O. It would be helpful to be able to 'step' the trace. Otherwise at 9600 baud, it is useless. It would be helpful to allow null input and be able to break out of the program. (On WANG, a halt step during input sets up a break on entering the return key.)

ISAM: ICS has developed an ISAM capability and is prepared to offer some BASIC subroutines that provide a reasonably comprehensive ISAM file maintenance sub-system to any application program.

VIDEOGIDE: ICS developed a package called VIDEOGIDE, two years ago, to help solve the problems that considerable volume of input created. VIDEOGIDE builds a protected screen face on a serial terminal. It currently runs on the LEAR SIEGLER ADM 3A, TELEVIDEO 912, HEZELTINE 1400 and HEATHKIT H-19.

NAMAD: ICS currently has a name and address update system (NAMAD) running on floppy disk that demonstrates VIDEOGIDE and the ISAM subroutines. We would be willing to send it to OSI for evaluation if requested.

John B. Christiansen
Jenkintown, PA

John:

Thanks for the comments. We would be glad to receive more information from you, particularly direct and detailed comparisons between your C-3-C, DEC, WANG, ALTAIR, etc. You are in a unique position to be able to make such comparisons.

Concerning software, if you would like to send us review copies, we will test and report on any items you would like to sell.

AL

* * * * *

LOAD

UP ON QUALITY SOFTWARE



That's right. The kind you get from SOFTWARE CONSULTANTS... it's efficient, cost-effective, and flexible. Next time you dump some old programs, remember to load up with the hardworking software you get from SOFTWARE CONSULTANTS.

We're an independent vendor dedicated to producing top quality software just for you, the OSI users, at the most reasonable prices around. As we've told you before, most of our products are modifications and/or extensions of existing OSI packages. Some of our items completely replace OSI's, and all of our products are easily extensible for your custom applications.

The many responses we've recently received have shown us just how badly many of you needed a quality source for OSI software. So, to keep up with your requests for more "good OSI stuff," we've been working overtime on lots of new products. We'll be telling you about each one of them as soon as they're ready. But in the meantime, we've gotta tell you our prices will be going up next month, so be sure and order now, before the increase.

We just don't have enough room in these ads to fully describe our products to you, so please call or write us for the latest copy of the free SOFTWARE CONSULTANTS product catalog. It'll give you all the facts on our current line and full details as each new product is introduced.

Remember, for fine OSI software at the most reasonable prices around, it's SOFTWARE CONSULTANTS. Take a quick look at this list and ask yourself...can you really afford to keep using any one else's OSI software?

1. OS-65D V3.2 DISASSEMBLY MANUAL

A super-complete manual that has it all -- 50 pgs. of disassembly listings, complete and clear comments on most every line, 10 pgs. of computer generated cross reference listings, and more! Praised by many OSlers who couldn't believe it 'til they bought one. A deal at \$24.95.

2. REF COMMAND UNDER BASIC

A complete, cross reference utility that'll find and list any BASIC line number, variable or numeric constant. It's available under 65D or 65U and comes on 5 1/4" or 8" floppies. This one will save your sanity, and cut out hours of wasted time. Yours for \$29.95.

3. SPOOLER/DESPOOLER UTILITY

A useful utility that feeds backed-up data to your printer for normal output, and leaves your screen free for other work by intercepting data bound for your printer and temporarily storing it on hard disk. Written in super fast machine language. Interfaces with serial and parallel printers.

4. FIG FORTH UNDER OS-65U

So far, the only one that's running under 65U at a price within reach. You get all the pluses, like terminal oriented editor, lots of printer and terminal tools, and more. Under multi-user, runs BASIC simultaneously, too. Unbelievable at \$79.95.

5. VIDEO ROUTINE

This convenient program really spices up your video system with little niceties like 24 separate control codes, horizontal and vertical plotting, and many variable screen parameters. Software extensions are available to connect this with the graphics resolution booster. The routine alone is \$24.95, with extensions, \$29.95.

6. GRAPHICS RESOLUTION BOOSTER

An ingenious piece of hardware that'll increase the graphic resolution per character by 8 times. With this device, your circles will be rounder and your curves smoother. A slick addition to your video-based system. Priced at \$49.95, or, with the video routine and extensions, \$79.95.

Unlike the majority of other software vendors, we offer our customers copies of source code (on floppies) for any of our products they've purchased. For a nominal fee of \$10, covering our material, postage, and handling, we'll send you the source code you choose. Simply write it in as you fill out the order coupon. Of course, dealer inquiries are invited for all of our products.



Enclose your check with coupon and mail to: SOFTWARE CONSULTANTS, 7053 Rose Trail, Memphis, TN 38134 USA. TN residents add 6% sales tax. Phone us at 901/377-3503 for rush orders.

Dear Software Consultants:

Sounds great. Here's my order. I want:

___ DISASSEMBLY MANUAL(S) @ \$24.95 ea
___ CROSS REFERENCE UTILITY(S) @ \$29.95 ea.
Disk size 5 1/4" _____ or 8" _____
___ OS-65D _____ or OS-65U _____
___ SPOOLER/DESPOOLER UTILITY(S) @ \$59.95 ea.
Std. parallel interface _____
Serial interface w/430 board or equivalent _____
Serial interface w/CA10X board _____
___ FIG FORTH(S) @ \$79.95

___ VIDEO ROUTINE(S) @ \$24.95 ea.
With software extensions @ 29.95
___ SOURCE CODE @ \$10.00 ea.
For _____
___ GRAPHIC RESOLUTION BOOSTER @ \$49.95 ea.
With Video Routine and extensions @ \$79.95 ea.
___ SOFTWARE CONSULTANTS Product Catalog Free

NAME _____
ADDRESS _____
FIRM _____
CITY/STATE _____ POSTAL CODE _____

PEEK (65)

The Unofficial OSI Users Journal

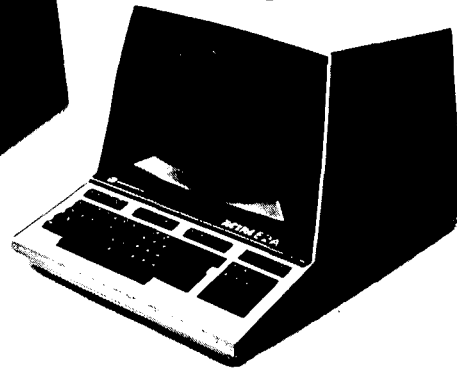
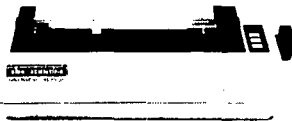
P.O. Box 347
Owings Mills, Md. 21117

BULK RATE
U.S. POSTAGE
PAID
Owings Mills, MD
PERMIT NO. 18

DELIVER TO:

DBMS, INC.

P.O. Box 347
Owings Mills, Md. 21117
(301) 363-3267



ACT-5A TERMINAL	Upper/lower case, 110-19,200 baud, unique split screen 39x48 character display, pass thru and buffered printer port, full cursor control and edit operations. 8 special function keys, reverse video, blinks and underlines fields, plus many other features	GRAY SCREEN \$829.00 GREEN SCREEN \$889.00
MIME-2A TERMINAL	Same features as ACT-5A but EMULATES HAZELTINE 1500 DEC-VT52...SOROC IQ-120	GRAY SCREEN \$829.00 GREEN SCREEN \$889.00
CAT MODEM	0-300 Baud, Acoustic, Answer/Originate	\$175.00
EPSOM MX-80	Matrix, Upper/Lower case printer	\$575.00

Order yours today and we will ship tomorrow freight collect.