PEEK (65)

The Unofficial OSI Users Journal

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

★★ $1.75 ★★

Editor: Al Peabody
Vol. 2, No. 9, Sept. 1981

INSIDE:

Column One

It was one of the shortest letters I have ever recieved, though I would hardly call it short and sweet! It said, "Sir: I hope to see more information, memory maps, decoded Basic operation Routine + other useful information, or this will be my last Renewal."

You fellows really know how to hurt a guy. Here I was thinking we had been **full** of information, memory maps and decoded Basic function Routine, and you, our readers, bring me up short.

Perhaps what our reader (who shall remain anonymous) means is, he wants to see more stuff he can use in HIS computer. Never mind all that fancy stuff about CP/M and Microspell and such. What about good old Basic, and the memory map for MY operating system. Excellent points.

All I can say is, coming up! A glance through this issue reveals some of the very goodies I hope he was talking about. A nice article on the Extended Monitor, with some suggestions for customization, if you are into tape systems and Assembler; an article (the first of a series) on 65D and all that goes with it, for minifloppy and floppy disk types. And a CP/M review and some sorely needed info on the C3 Utilities for the C3 user. Seems to me we have about all the bases covered, from C1P to C3B.

Even the ads this month are exciting. How about double density floppies for starters? Or double speed Basic?

So I hope you will all stick with us. Because remember, I write maybe 20% of what goes into this magazine. You guys write the other 80%. It is your journal, and you determine what we print, both by comments like this month's stopper letter, and by what you send in, as letters or articles. So do let me know what you think of what you see, and what you would like to see more of.

Last month, I reported that our CBBS had fallen on hard times, but was up and running. About the time that issue hit the mails, would you believe we were struck by lightning?!? As of today the board is working fine, but our modem is off to the modemfixer store for a week or so to replace some melted parts. We are using the time this gives us to insert some new goodies, new commands, new unique message numbering and a surprise or two. Meanwhile, since there is no modem, we won't be answering the phone, so at least you won't be paying for phone calls when you didn't get to work with the board. Have no fear, we will be back better than ever!

There are several product and service announcements in this number. We have agonized all out of proportion to their significance about what to do with the news releases we receive. They are really very thinly disguised ads, often running to a full page or more, for new products, programs, etc. We don't feel it is really fair to our regular advertisers, who pay good bucks to tell you what they are offering, to just print these freebies without restraint. On the other hand, you want to know what's new, and this stuff is by definition new. So we have evolved the following policy:
We will print new product and service announcements one time, edited as we see fit (cut severely if too long), as soon as we have room for them, in priority of apparent interest to our readers (announcements of new products not usable on OSI gear will not be printed).

The boolean logic contest goes on. An entry is printed this month, two others have been received over the CBBS and will be printed next month. We need more. Remember, the object of the contest is to use a statement like

```
100 Y=A*(D>9)+3
```

in a BASIC program, making use of the fact that the expression in parentheses is evaluated differently depending on whether it is true or not. (See last month's Column One for more details.) Best use wins instant fame and a rather worthless prize.

al

THE BEGINNING ASSEMBLER

PART III

by Al Peabody

In the June 1981 issue, I reported what I had learned about assembling programs under 65D, then bringing them over into 65U to use with BASIC programs. All this was in theory, of course. I had never actually done it with a "real" program, one which does something useful. Now I have!

As regular readers know, I am very much interested in data communications by computer. However, up to now I have been somewhat frustrated in this area. My computer is a C3-D, with Shugart's neat 10 Mbyte hard disk. This means I rarely use floppies at all. I boot up on the hard disk, the first program which runs on bootup contains the control codes I need to set my terminal to the condition I want, the computer is actually located in a closet 10 feet or more from the terminal -- you get the picture. Dealing with floppies is a nuisance.

However, when I first started working with Micronet and the PEEK(65) CBBS, I had to do more than deal with floppies. I had to unplug the terminal from the computer, plug it into the modem, adjust the terminal's baud rate to 300, set it on half duplex to allow me to type in the control codes for full brightness, reset full duplex to work with the remote computer, and hit the "break" key on the terminal before I was finally ready to dial the phone. Clearly a terminal emulator program was needed.

Pretty soon, Phil Lindquist came to the rescue with SMARTER-M, a neat emulator with all kinds of special features like storage of incoming data

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

```
10; Dumb terminal emulator..."Ping-pongs" back and forth
20;  between CA-10 port one and serial terminal port, and
30;  sends whatever it finds on one to the other
40;
50;  Can be inserted into 65U Basic programs with LOAD 48
60;
70;       by Al Peabody    June 1981
80;
90;
100          TERMST=$FC00         ;TERMINAL STATUS BYTE
110          TERMDA=$FC01         ;TERMINAL DATA PORT
120           MODST=$CF00         ;MODEM STATUS BYTE
130           MODDA=$CF01         ;MODEM DATA PORT
140           OUTCH=$FE0B         ;PRINT CHR ROUTINE IN MONITOR ROM
150            INCH=$FE00         ;GET CHR FROM TERMINAL
160          ACIAMR=%00000011     ;MASTER RESET OF 550 BD ACIA
170          ACIAST=$11           ;RICK W SAYS THIS IS RIGHT
180;
190               *=$6000             ;BEGIN AT START OF BASIC
200;
210    INIT    LDA    #ACIAMR         ACIA MASTER RESET
220            STA    MODST
230            LDA    #ACIAST         SET ACIA STATUS
240            STA    MODST
250            LDA    #126            SET HAZ1420 TO BRIGHT
260            JSR    TOUT
270            LDA    #31
280            JSR    TOUT
290            JSR    QA              HALF OR FULL DUPLEX?
300            JSR    PRNT            PRINT RETURN MESSAGE
310    TERM    LDA    TERMST          CHECK TERM STATUS
320            LSR    A               CHECK FOR BYTE
330            BCC    MOD             NONE, GO TO MODEM
340            LDA    TERMDA          BYTE THERE
350            AND    #$7F            MASK PARITY BIT
360            CMP    #04             IS IT CONTROL D?
370            BEQ    END             YES, RETURN TO BASIC
380            JSR    MOUT            SEND TO MODEM
390            TAX                    SAVE CHAR
400            LDA    HAFF            HALF DUPLEX?
410            CMP    #'H
420            BNE    MOD             NO, CHECK MODEM
430            TXA                    GET CHAR BACK
440            JSR    TOUT            SEND TO TERM
450    ISCR    CMP    #$0D            IS IT A CR?
460            BNE    MOD             NO CHECK MODEM
470            JSR    CRLF            PRINT A LINE FEED
480    MOD     LDA    MODST           CHECK MODEM STATUS
490            LSR    A               CHECK FOR BYTE
500            BCC    TERM            NONE, GO TO TERM
510            LDA    MODDA           BYTE THERE
520            AND    #$7F            MASK PARITY BIT
530            JSR    TOUT            SEND TO TERM
540            TAX                    SAVE CHAR
550            LDA    HAFF            HALF DUPLEX?
560            CMP    #'H
570            BNE    BACK            NO, RETURN TO TERM
580            TXA                   GET CHAR BACK
590            JSR    MOUT           SEND TO MODEM
CMP    #$0D           IS IT A CR?
610            BNE    BACK           NO, RETURN TO TERM
620            JSR    CRLF           PRINT LF
630    BACK    JMP    TERM           CHECK TERM
640    TOUT    PHA                   SEND CHAR TO TERMINAL
650            LDA    TERMST         CHECK STATUS
660            AND    #02            READY FOR A BYTE?
670            BEQ    TOUT+1         NO, WAIT
680            PLA                   GET CHAR BACK
690            STA    TERMDA         AND PRINT IT
700            RTS
710    END     LDA     #'^            PRINT '^B' TO TERMINAL
720            JSR    TOUT
730            LDA    #'B
740            JSR    TOUT
742            LDA    #0              RETURN WITH Y=0
744            TAY
750            JMP    (8)             AND RETURN TO BASIC
760    MOUT    PHA                    SAVE CHAR
770            LDA    MODST           CHECK MODEM STATUS
780            AND    #2              READY FOR BYTE?
790            BEQ    MOUT+1          NO, WAIT
```

continued

GOODIES for OSI Users!

from

master charge
THE INTERBANK CARD

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

VISA

( ) **C1P Sams Photo-Facts Manual.** Complete schematics, scope waveforms and board photos. All you need to be a C1P or SII Wizard, just $7.95 $ ____________

( ) **C4P Sams Photo-Facts Manual.** Includes pinouts, photos, schematics for the 502, 505, 527, 540 and 542 boards. A bargain at $15.95 $ ____________

( ) **C2/C3 Sams Photo-Facts Manual.** The facts you need to repair the larger OSI computers. Fat with useful information, but just $34.95 $ ____________

( ) **OSI's Small Systems Journals.** The complete set, July 1977 through April 1978, bound and reproduced by PEEK (65). Full set only $15.00 $ ____________

( ) **Terminal Extensions Package** - lets you program like the mini-users do, with direct cursor positioning, mnemonics and a number formatting function much more powerful than a mere "print using." Requires 65U. $50.00 $ ____________

( ) **RESEQ** - BASIC program resequencer plus much more. Global changes, tables of bad references, GOSUBs & GOTOs, variables by line number, resequences parts of programs or entire programs, handles line 50000 trap. Best debug tool I've seen. MACHINE LANGUAGE - VERY FAST! Requires 65U. Manual & samples only, $5.00 Everything for $50.00 $ ____________

( ) **Sanders Machine Language Sort/Merge** for OS-65U. Complete disk sort and merge, documentation shows you how to call from any BASIC program on any disk and return it or any other BASIC program on any disk, floppy or hard. Most versatile disk sort yet. Will run under LEVEL I, II, or III. It should cost more but Sanders says, "...sell it for just..." $89.00 $ ____________

( ) **KYUTIL** - The ultimate OS-DMS keyfile utility package. This implementation of Sander's SORT/MERGE creates, loads and sorts multiple-field, conditionally loaded keyfiles. KYUTIL will load and sort a keyfile of over 15000 ZIP codes in under three hours. Never sort another Master File. $100.00 ____________

TOTAL	$ ________
MD Residents add 5% Tax	$ ________
C.O.D. orders add $1.40	$ ________
Postage & Handling	$ 3.00
TOTAL DUE	$ ________

( ) Cash enclosed ( ) Master Charge ( ) VISA
( ) C.O.D., add $1.40

Account No. ____________________ Expiration Date ____________

Signature ______________________________

Name ______________________________

Street ______________________________

City ____________________ State ____________ Zip ________ in a RAM buffer if desired, storage of the buffer on disk, etc. -- but it was written in 65D, so I was back to floppies, and a system I was not very familiar with.

Then I found the source code for a terminal emulator program of somewhat less sophistication in another OSI publication, the newsletter of OSIO. Though designed for 65D, the program was so simple and well documented, even I could understand it, and with some modification, it seemed to me it would work just fine under 65U, "my" system. So the experiment was on!

First I typed in the source code for the program, using the 65D assembler and changing those parts which were different under 65D and 65U (for example, the original program returned to BASIC by jumping to $1B25; I returned by loading the Y register with a flag byte indicating whether everything was okay, then jumping indirect through $08, as I explained last month. Also, of course, the origin of the program had to be $6000, the start of BASIC's workspace under 65U).

After making these changes and checking the listing for any visible errors, I followed the steps I outlined last month to bring the machine code over into 65U. The BASIC program I wrote to test it was dirt simple:

```
10 PRINT"TERMINAL EMULATOR"
20 INPUT"OKAY";QA$
30  IF  LEFT$(QA$,1)="N"
   THEN RUN"MENU
40 PRINT"CTRL-D TO
   RETURN TO BASIC"
50 POKE 8778,96:
   POKE 8779,0
   Y=USR(X)
60 PRINT Y
```

Now, answering anything which doesn't begin with N to the question "OKAY?" will put me in the terminal mode. Since the program is set up so that a control-D returns to BASIC, I can get back anytime. As of now, all that will happen is I will see the value of Y, the number I loaded into the Y register in the terminal program before jumping back to BASIC, plus the OK.

But of course, I have plans for the future. I intend to use this program in sort of the reverse of the usual manner. Usually, we write the bulk of our programs in BASIC, then call machine language routines as needed to do things for which BASIC is not well suited. I plan to do the reverse. The bulk of the program is the terminal emulator, written in ML; BASIC will be called from time to time to do stuff which is so hard to do in assembler it makes my head hurt to think about it -- stuff like putting parts of files away on disk while keeping track of where we are in the file, calling up parts of files for transmission to the other computer, etc. I will keep you posted as we go along.

You will notice that this program uses NO protocol. It is the essence of the "dumb" terminal, just sending whatever you type out over the line and putting onto the screen whatever comes in over the line. As we further develop the OSI protocol, with your help and suggestions, I will incorporate it fully into the growing program.

```
800            PLA .                    GET CHAR BACK
810            STA    MODDA             AND SEND TO MODEM
820            RTS
830    CRLF    LDA    #$0A              LF FOR NON-AUTO CRLF TERMS
840            JSR    MOUT              SEND TO MODEM
850            JSR    TOUT              AND TO TERM
860            RTS
870   QUEST    .BYTE  '(H)alf or (F)ull duplex?'
880    QA      LDX    #0                ASK QUESTION
890            LDA    #$0D              FIRST CR
900            JSR    TOUT
910            JSR    CRLF              AND LINE FEED
920    LP3     LDA    QUEST,X           GET A BYTE OF QUESTION
930            JSR    TOUT              PRINT TO TERM
940            CMP    #'?               LAST CHARACTER OF QUESTION?
950            BEQ    ANS               THEN GET ANSWER
960            INX                      NO,
970            JMP    LP3               GET NEXT CHAR
980    ANS     JSR    INCH              GET ANSWER
990            CMP    #'H               IS IT H?
1000            BEQ    STUFF             SAVE IT
1010            CMP    #'F               OR F?
1020            BNE    QA                NEITHER, ASK AGAIN
1030   STUFF    STA    HAFF              SAVE ANSWER
1040            LDA    #$0D              PRINT CR
1050            JSR    TOUT              TO TERM
1060            JSR    CRLF              AND LINE FEED
1070            RTS
1080   HAFF     NOP                      SPACE FOR ANSWER (H/F)
1090   MESG    .BYTE   'Type Ctrl-D to return to BASIC. '
1100   PRNT     LDX    #0                PRINT MESSAGE
1110            LDA    #$0D              FIRST A CR
1120            JSR    TOUT              TO TERM
1130            JSR    CRLF              AND AN LF
1140    LP4     LDA    MESG,X            GET A CHAR
1150            JSR    TOUT              SEND TO TERM
1160            CMP    #'.               LAST CHAR?
1170            BEQ    RET               THEN RETURN
1180            INX                      NO, GET NEXT
1190            JMP    LP4
1200    RET     RTS
```

★ ★ ★

A BUG FIXED
PEEKer FIXES WORDSTAR
by Al Peabody

A couple of months ago, I reported on WordStar, the famous word processing system. One of the things I reported was that for all its brilliance, WordStar was s-l-o-w. If you type at anything like normal speed, I said, you would get ahead of it and it would drop letters when it went away to do things like wordwrap and screen formatting. I could hardly believe it, but there it was.

Well, it no longer is. A PEEK(65) reader, Al Black, called us on the phone to report that he had previously had the same problem but, with the assistance of MicroPro International, WordStar's vendor, had fixed it. I have now installed the same fix on the WordStar which we are just about to return to Lifeboat (we only had it on loan), and it WORKS. I can type as fast as my little fingers will fly, and WordStar never drops a letter.

For any of you who might get WordStar (and for dealers who sell it), here is the fix, how it works, and why:

The fix--
Insert your Wordstar disk and boot up. Then type INSTALL<CR> and select option D, reinstall an existing Wordstar File; WS.COM, or whatever other name you might have given it. Run through all the selections of terminals, etc., picking U for unchanged in each case, then answer N to the question as to whether patches are complete. INSTALL will now ask you what address you want to change, and you reply INISUB: <CR> (don't forget the colon). By succesively hitting the CR, you will be led along to locations INISUB:+1 and so forth, and you can insert the following routine:

```
C3 F7 02 C3 03 03
```

What you have done is to install two JMP instructions, one to be performed just as you start working with WordStar, another just as you finish. Now type 02F7 as the next location to be changed, and insert the following routine at that point, one byte at a time:

```
3E 40 32 03 00 3E 00 32 17
BF C9 00 3E 01 32 17 BF C9
```

These two routines do a couple of things on startup. They insert a 40 in location 0003, the I/O status byte for CP/M and then put a 00 in location BF17. If your particular combination of boards needs an 80 in location 003, then you will start off with 3E 80. On the way out of Wordstar, the second routine puts a 01 in BF17.

With this fix in place, you can type even a nonsense sequence of letters like BNB BNB BNB (I always find I can type nonsense faster than anything else), and no matter how fast you do it, your WordStar won't miss a trick.
Here's how it works, courtesy Al Black:

BF17 is the lowbyte of the address of the terminal data port (as it was originally). What is happening here is this. As WordStar is printing stuff out to the screen, for example as it is reforming a line after wordwrap, between each character it checks to see if anything has come in on the terminal. In fact, it does so twice. Once to see if you have typed in another character, so that it won't lose characters while it is reformatting lines and stuff; another time just to see if you have typed in a control S or control C.

Here's the rub. When it checks for control S or C, it checks the status port, finds that something is there, then reads it out of the data port and has a look to see if it is a control S or C. But when it reads the byte out of the data port, it automatically resets the status port! So, while WordStar is reformatting a line, you type in another character. WordStar checks for control S or C, erases your character, then goes on with the formatting! A fraction of a second later, when it checks to see if you have typed another data character in, whattayaknow, there is no character there.

So all the fix does is tell WordStar to look in the status port for control S or C, not in the data port. The probability of finding a control S or C in the status port is negligible, so it doesn't erase the characters from the data port. Hence, they are stored away in a buffer and are available to print out the Bug fixed. Word processor now even more brilliant than before. Thanks, Al Black.

★

CP/M FOR OSI

* "The Software Bus" review *

by Al Peabody

CP/M is universal; CP/M is on everybody's computer; CP/M is the national standard; CP/M is clunky.

What?! CP/M, the system which thousands use, hundreds write programs for, millions would like to have -- clunky??

Yep, clunky, bunky. Let me explain. I was intorduced to micros on OS-65U. Yes, I know, OS-65U has its faults. There are at least three different versions all labeled 65U V1.2. You are pretty much forced to work in BASIC or 6502 Assembler. The factory has kept the source code a deep, dark secret, to frustrate those of us who would like to improve the breed, or at last customize it. But consider this. When I want to load or save a file on my floppy drive in OS-65U, the drive goes "click, buzz" and a 20K file is ransferred to or from disk so fast that I actually had to do a poke to the OS to slow down the stepping rate -- it was too much for the poor disk drive.

By way of contrast, if I want to load or save a file under CP/M, the drive head first s-l-o-w-l-y steps in to track 3 or so, then clicks once, reading out what is called the file control block of the file in question, then s-l-o-w-l-y steps out to wherever the file starts, then clicks two or three times and reads in -- 128 bytes! Then it s-l-o-w-l-y grinds its way back to the file control block on track 3, then s-l-o-w-l-y chatters out to wherever the next 128 bytes of my file is and takes them in. And so forth. Wait: there's more. "Erasing" a file merely loses its File Control Block. Nothing at all happens to the space on the disk where the data was. Oh yes, something does happen. CP/M now knows those sectors are available for re-use. So the next time I create a file, they will be filled first. Only if my new file is larger than the one erased, after those sectors are filled, CP/M is smart enough to go to the next empty sector. This is called "dynamic file allocation." It has its advantages.

You don't need to "create" a file before you can put stuff into it. All that is taken care of for you. You never have to "repack" your disk like you periodically have to do under OS-65U. But what a price you pay! After a disk has been in use for a while, you type whatever you need to type to save a sizeable file or load a large program, then go take a coffee break while the disk drive clunks and buzzes around. In one article I read recently, the author was bragging that his CP/M computer could load a 15K program in "only" 20 seconds, instead of the usual 45!

Okay, then, smart guy, if CP/M is so awful, why are you writing this series of reviews of CP/M products; and why are you writing it with WordStar, a CP/M word processor? Now we get down to the essence. Like most businessmen, housewives and non-hackers in general, I am not really a computer programmer. I am what you call a computer user. Big difference. Sure, I do some programming (don't we all?). But if I had to make my living as a programmer, it would get real hungry around my house. What I need is applications software which is already written and running. Accounting systems. Good word processors. Terminal emulators. Lots more. And boy, does CP/M have it! Just looking through the Lifeboat catalog makes me drool. And with all those CP/M systems out there in computerland (now including every OSI C3), more is coming every day. "The software bus," they call it. And with reason. I might wish it had OS-65U's byte-addressable file structure and rapid disk access. I might never want to have to figure out how to write programs to read and write 128 bytes at a time to or from disk and still choose the record length I want to use. But I don't have to. There are 3,000 CP/M programmers out there doing it for me. All I have to do is buy the software and enjoy it!

Well, that isn't quite all, which brings us back to Lifeboat again. As most PEEKers know, there are several stages involved in using a program:

1) Buy the software, run home with a grin on your face, and install it on your computer.

2) Try to make it work.

3) Read what passes for a manual 6 or 7 times.

4) Curse the vendor a few times, wish dire things upon him and all his tribe, and try to find somebody (dealer, other user, computer clubber) who knows how to make it work.

What I am saying is, with CP/M, steps 1) through 3) are still pretty much the same, but step 4) is easy. IF you bought your CP/M product from Lifeboat Associates, call them on the phone, any time between 11 AM and 7 PM New York time, and ask to speak to an engineer. I have done this several times, with questions ranging from simple to really tough, and those guys really stuck with me. Once my question just stumped the fellow, and he took my number and, honest, called me back the same day with the answer! Ever hear of OSI or Apple or Pet doing that? Or of a cut-rate CP/M mail order house doing that? Me neither.

A note about versions. Until very recently, all OSI dealers could offer you was OS-CP/M version 1.4, which had serious problems. Mainly, every time CP/M wanted to get one of those 128-byte sectors off the disk, OS-CP/M (get this) switched processors back to the 6502, read a track off the disk, essentially in 65D, then switched back to the Z-80 and CP/M. If you think it's slow reading data 128 bytes at a time, try reading data 3584 bytes at a time with one chip, sticking it in a buffer, cal-

ling another chip, taking 128 bytes out of the buffer, then switching back again and...
Now, however, both Lifeboat and your OSI dealer have CP/M 2.2 for OSI. All I/O is handled by the Z-80. Also it has the capacity to work with OSI's big hard disks. I am told by the end of the summer we will have a "mini-winnie" 5.25" winchester with 10 Mbytes capacity, and with OS-65U, OS-65D and CP/M resident on the disk at the same time. With a national service network and source code available for all 3 systems. I can hardly wait!

So, where are we? Seems that CP/M is a clunky operating system, primarily due to slow disk access and difficult random-access file handling techniques (might be much better on hard disk -- PEEKers who know, please write in), with loads and loads of neat software available, and plenty of technical help readily at hand, IF you got your stuff from Lifeboat (or maybe a few others -- again, PEEKers who know share your info).
If you primarily want to program in BASIC, or play games, or just have fun with your computer, maybe you better skip CP/M. But if you want to do some serious business computing with minimal hassle and maximum future prospects for further expansion and improvement of your system, be glad your C3 will run CP/M! For all its faults, it is indeed the software bus.

★

Future Now In the Don't-Look -Now-Here-Comes-The-Future department:

John Brunner, in the SF novel SHOCKWAVE RIDER, postulates a society bound together by ubiquitous video terminals acting as communication devices and data terminals entering onto a massive data net. As a primary plot device he describes a natural catastrophe (earthquake) that leads to the creation of an organization called the ten 9's. The 10 9's refers to the phone number the organization can be reached at; the only function of the people at the answering end of the phone is to LISTEN. Period. No judgements, no comments, nothing except for a final 'I heard that' to signify that someone was listening. The organization is used as a sort of antiseptal Father confessor. In the latest issue of OMNI (July 1981), the 'MIND' column reports on a Lower Manhattan artist's creation: a telephone answering machine used for catharsis. 'Chris' (not his real name, according to OMNI) originally thought of the set-up as a means of getting tapes of criminals confessing to their crimes, which he would then replay in a Soho gallery. After plastering Times Square with posters stating 'You have wronged people. It is the people you must apologize to, not to the State, not to God. Get Your Misdeeds Off Your Chest! Call (212) 255-2748'. What he got was people from all over the States calling in and using the system as a confessional.

My, my, it was only a science fiction book, too...

I strongly urge you read both SHOCKWAVE RIDER and the July 1981 OMNI, preferably in that order. Interesting reading all the way around.

++hobbitt

★

OS65D3 #1 IN A SERIES

'D.R. "STRETCH" Manley
5664 E. Evans Creek Road
Rogue River, OR 97537

Well, you see what happens when you volunteer to do something? I told Al that I'd write a column on OS65D3, if he wished, and he took me up on it. Here goes...

****BUGS AND FIXES****

For starters, we ought to fix our systems so that they work right. O.S.I. has published a few fixes for bugs in the '65D system, so in case you haven't heard of them, or you don't have a dealer nearby, here are the ones found in the technical newsletters (T.N. in the following text).

#1, from T.N. #3, pages 8-12 (numbered 1-5 in T.N.) For mini-floppy systems only!!!! (5 1/4" disks)
This fix is supposed to cure drive to drive incompatibilities. Since I have 8" drives, I don't know if this is a good fix or not. Caveat emptor.

Two locations need to be changed, one in the operating system, and one in the diskette utilities. If you have never used the track zero read/write utility before, now is the time to learn. Put a diskette with the operating system on it in drive A and boot up. Enter "UNLOCK" to get to BASIC's immediate mode. Then type "EXIT" to get to the operating system kernel. If you get an "A*" for a prompt from the operating system. Type "EM" to call the extended monitor. Monitor's prompt is a ":". Type "!CA 0200=13,1" to tell the EM to tell the disk system to call track 13, sector 1 from disk to memory location $0200. Now the diskette utility is in memory at its normal run time location. We'll change it first, then use it to change the operating system. Type "@0638". ("@" is shift-p or shift-O). You should get "0638/01" displayed. If not, skip the rest of the fix. Type "0A" to install the fix. Type "!SA 13,1=0200/5" to save the fixed utility back to disk. Now to change the operating system itself. Type "EXIT" to get back to the kernel. Type "GO 0200" to run the utility. Type "2" to select the track zero read/write part of the diskette utility. Type "R4200" to read the contents of track zero to memory locations $4200 and up. This is above the extended monitor and operating system, and exactly $2000 bytes higher than it would be if it had been booted in normally. Type "E" to exit the utility. Type "RE EM" to re-enter the extended monitor. Type "@47A0". If "47A0/01" is not displayed, go no further with this fix. Type "0A" to enter the fix. Type "EXIT". Type "GO 0200". Type "2" to select the track zero read/write utility again. Type "W4200/2200,8". This saves the memory from $4200 for 8 pages (at 256 bytes per page) to track zero, and puts a load vector in front of $2200. That is so the boot routine knows where to write the stuff from track zero when it reads it in.

#2, from T.N. #3, pages 13 & 14. This fix applies to both 5 1/4" and 8" disk systems. I found that this fix had been installed on my diskette at the factory, but it was not quite at the place specified in the T.N. there was a difference of one byte in the location. Let's check yours and see where it's at. Boot up again, type "UNLOCK" and "EXIT" to get to the kernel. Type "EM" and then "EXIT". This sets the extended monitor up. Type "CA 0200=13,1" for 5 1/4" systems, or "CA 0200=01,2" for 8" systems. Type "GO 0200" for both systems. Type "2" for the track zero utility. Type "R4200" and "E" to read track zero in and exit to the kernel. Type "RE EM" to re-enter the extended monitor. Type "@4886". You should see "4886/13" displayed. If you see "4886/28" then the fix has been installed. If it was "13", then a little further checking is in order. This is where I found the difference on my disks. Type "@4898". If you see "4898/D0 then you need to install the fix. If you see "4898/09" then you probably have the same system I do. If you have "4898/D0" then enter "4C" and a line feed (LF). You'll get "4899/6F". Enter "09"(LF). You'll see "489A/60". Type "28" and a carriage return (CR). Now type "@4886(CR)" again, and when you get "4886/13", enter "28(CR)". That should fix it. If you got "4898/09", then hit (CR), and enter "@4897(CR)". If you get "4897/C4", hit (LF) twice. You should see "4898/09" and "4899/28". If you do, the fix is installed, and you can skip the rest of this one. If you don't, then all bets are off. You got one I don't know about. If you had to install the fix, you had better save it to disk, right? Type "EXIT" and "GO 0200" to get back to the utility. Type "2" to select track zero READ/WRITE again. Type "W4200/2200,8" for both 5 1/4" and 8" systems. Your fix is saved.

#3. This fix is probably the most important. It's from T.N. #3, page 15. If you have been using random access files, then you might have run into this problem before. It's a dilly. I lost a lot of hair over this one. The problem manifests itself only if you try to access a record with a "DISK GET, NN", where NN is 384 or more. Since I needed files with record numbers of 1000 and bigger, this one hurt. It seems that the routine used to figure the relative track of the file (relative to the first track of the file) Lost its mind after record #383 and gave a "FC ERROR", which you would normally only get if you tried to access a record past the end of the file. We don't have to worry about the track zero utility on this one, since it is in a sector of 1 page that loads into the disk directory buffer after a BASIC 'DISK OPEN, N, "filename"' has been executed. Saves memory that way, since the directory buffer isn't being used after the open, anyway. Boot up again, and type "UNLOCK" and "EXIT". Type "EM". Type "CA 4E79=12,4" for 5 1/4" systems, or "CA 4E79=08,4" for 8" systems. This puts the code into the directory buffer where it would normally be. Type "@4F00". You should see "4F00/30". Type line feed

OSI AARDVARK OSI
NOW MEANS BUSINESS!

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 capabibilities. 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

NEW-NEW—NEW
TINY COMPILER

The easy way to speed in your programs. The tiny compiler lets you write and debug your program in Basic and then automatically compiles a Machine Code version that runs from 50-150 times faster. The tiny compiler generates relocatable, native, transportable machine code that can be run on any 6502 system.

It does have some limitations. It is memory hungry — 8K is the minimum sized system that can run the Compiler. It also handles only a limited subset of Basic — about 20 keywords including FOR, NEXT, IF THEN, GOSUB, GOTO, RETURN, END, STOP, USR(X), PEEK, POKE, -,=,*,/, (,),<>,Variable names A-Z, and Integer Numbers from 0-64K.

TINY COMPILER is written in Basic. It can be modified and augmented by the user. It comes with a 20 page manual.
TINY COMPILER — $19.95 on tape or disk

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!

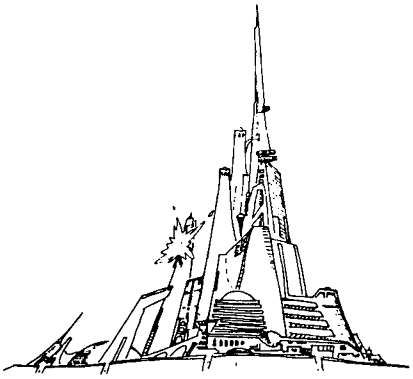
SUPERDISK II

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 âliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. Specify system — A bargain at $9.95

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.

NIGHT RIDER - You've seen similar games in the arcades. You see a winding twisting road ahead as you try to make time and stay on the road. NIGHT RIDER uses machine code to generate excellent high speed graphics - by the same author as MINOS.
NIGHT RIDER — $12.95 cassette only

THIEF - Another machine code goody for the C1P cassette only. You must use mobile cannon to protect the valuable jewels in the middle of the screen from increasingly nasty and trigger happy thiefs. Fast action and fun for one or two players. THIEF $13.95 on C1 cassette only!

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

(LF). You should see "4F01/16". If you see "4F01/65" then the fix has already been installed, and you can skip the rest of this one. If you see "4F01/16" then follow the next steps carefully. A lot of code has to be changed to get this to work right.

```
YOU SEE:        YOU ENTER:
4F01/16         65(CR)
:               @4F18(CR)
4F18/AD         EA(LF)
4F19/92         EA(LF)
4F1A/2F         EA(LF)
4F1B/18         F8(LF)
4F1C/F8         18(CR)
:               @4F67(CR)
4F67/00         AD(LF)
4F68/00         92(LF)
4F69/00         2F(LF)
4F6A/00         F0(LF)
4F6B/00         AF(LF)
4F6C/00         F8(LF)
4F6D/00         18(LF)
4F6E/00         AA(LF)
4F6F/00         A9(LF)
4F70/00         (LF)
4F71/00         69(LF)
4F72/00         01(LF)
4F73/00         CA(LF)
4F74/00         D0(LF)
4F75/00         FB(LF)
4F76/00         F0(LF)
4F77/00         A4(CR)
:               EXIT(CR)
```

Now to save it to disk. Type "SAVE 08,4=4E79/1" for 8" systems, or "SAVE 12,4=4E79/1" for 5 1/4" systems.

done

#4. This one, from T.N. #4, pages 5 & 6, is fairly simple. I'm proud to say that I found it on my own, before I acquired a complete set of the T.N. Now is the time for you to do one on your own. The place to be fixed is one byte, located at $24BA normally. (If you call track zero in to $4200, it will be at $44BA). It is $4F, but should be $4D. That's all to that one, but if you have ever tried to input from device #8 (the CA-10-X or 550 board), you know that it doesn't work without this fix. I will (eventually) probably hook up a modem to my CA-10-4 board, so I know the operating system is ready.

That wasn't so hard, was it? I found that with a little practice, and a disk that I use just for experimenting on, fiddling with the existing system can be fun and instructive.

#5. Here's one for you folks that are using '65D3.1, the home control system. This one is simplicity, itself. It's all done in BASIC! From T.N. #8, page 3. If your real time clock runs slow when you do a lot of remote control, put these into effect. This must be done for every program that does remote control.

1. Load the program that is set up for A.C. remote control with a 'DISK!"LOAD filename".

2. In the immediate mode, if your program has: 0 file buffers, type in "POKE13053 ,32: POKE13054,30: POKE13055 ,50(CR)"

3. Put the program back to disk with a 'DISK!"PUT filename"'. That's all for that one. Remember to do it for every program that uses A.C. remote control.

#6 is for you folks with the C1P machine only. From T.N. #10, page 11. If your assembler "hangs up", this is the medicine for it. Call the extended monitor, (which also automatically loads the assembler, in case you didn't know), and type "@1563".

```
YOU SHOULD SEE:        ENTER:

1563/6D                9F(LF)
1564/15                24(CR)
:                    EXIT(CR)
A*               SA 09,1=1200/5
```

And the fixed part of the assembler is saved.

#7 Here's a fix aimed at the '65D3 crowd, and WP-2 also. Yep, WP-2 uses OS65D3 as a base. The word processor is in the place where the ASM/EM normally resides, and the utilities are all in BASIC. Here's what the fix does for you. If you are printing to more than one device at a time (by POKEing 8994 to set more than 1 bit, for example), and you hit control-C, when you type "CONT" to resume, you'll not get output to the high numbered device, only the default or low device. The system resets the input and output device to the default device on a control-C. This code will let your "CONT" resume with the same output as before. I didn't implement this, as I want control-C to stop everything but the default device. However, here's how it's done...

Boot up with a regular OS65D disk, and get to the operating system kernel (A* prompt). Call the extended monitor in, then the diskette utility. If you are going to change a WP-2 disk, insert it now. "GO 0200". We want the track zero utility, and we will read it into $4200, as usual. "REEM". This is a list of the locations, with their current and desired contents. You should be an old hand at this by now!

✓ done

LOCATION AND CONTENTS	DESIRED
4339/AD	A0
433A/21	00
433B/23	AD
433C/A0	21
433D/00	23
433E/F0	D0
434D/4A	D0
434E/E8	22
434F/90	E8
4350/09	4A
4351/48	90
4352/8A	09
4353/48	48
4354/20	8A
4355/71	48
4356/23	20
4357/68	76
4358/AA	23
4359/68	68
435A/E0	AA
435B/07	68
435C/D0	D0
435D/EF	F1
4371/0A	8C
4372/8D	78
4373/78	23
4374/23	D0
4375/98	D9
4376/18	0A

Save track zero back to disk with the disk utility (W4200/2200,8), and this one's done.

There is one more change, but it is very long and involved. It concerns the stepping rate problem encountered with the Siemens 8" drives, series "D". I have found that OS65D3 works fine with no changes on the one set of "D" drives I've tried it on. However, the same set wouldn't run OS65U until the stepping rate was changed. If you have a set of the fast drives ("D" series), see your dealer about getting T.N. #23. It has the changes for OS65D, 65U, and WP-2 to make the step rate adapt to the drives, automatically.

Well, I've run down like a dollar watch, so 'bye 'til next time, when we look at a few tricks and secrets of OS65D3.

STRETCH

* * * * *

Editors note: see T.N. #30 for additional stepping rate changes to OS65U.

★

NOTES ON THE OSI EXTENDED MONITOR

by Kerry Lourash
1220 North Dennis
Decatur, IL 62522

Here's an explanation of the innards of the OSI X Mon. (cassette version) and some tips for customizing and improving it.

When I bought the X Mon., I must confess I didn't know what I was buying. This was due both to my ignorance of machine language and OSI's sketchy description of its product. The X Mon. is a handy tool, even if you only dabble in machine language. Here are some of the things it can do:

DISPLAY - displays memory locations as hex or graphic characters, changes contents of memory, looks forward or backward through memory, and, best of all, lets you see the contents of 25-30 memory locations at a time when your entering ML code.

DUMP - displays a specified block of memory contents. Handy if you're looking for one byte in a haystack or want to see a program's effect on a block of memory.

FILL - fills a specified block of memory with a value (1-255). useful, with the DUMP command, to check a program's effect on memory.

MOVE - moves a specified block of memory to another part of memory. Used with the RELOCATE command to move ML programs.

RELOCATE - similar to the MOVE command , but it also adjusts all addresses that refer to locations within the block moved.

DISASSEMBLE - converts memory contents to ML mnemonics. Displays 24 lines of code, with a one-key repeat that lets you skim through a program. An invaluable command.

SEARCH - searches a specified range of memory for a sequence of 1-8 data bytes or ASCII characters. Useful, but finds only the first occurrence of the object of search. Doesn't allow "don't care" bytes in object of search.

BREAKPOINTS - allows insertion of up to eight breakpoints to troubleshoot ML programs. A breakpoint stops program execution at a specified address and prints contents of A,X,Y, status registers and stack pointer. Program execution can be resumed after breakpoint is encountered.

SAVE - saves specified block of memory to tape in checksum format.

LOAD - loads data in checksum format from tape.

VIEW - allows tape contents to be viewed on TV without loading into memory. A useful feature that BASIC should have.

CALCULATE - sum, difference, product, or quotient of two four-digit hex numbers. A hex-decimal, decimal-hex converter would be of greater utility.

Even if you don't do a lot of ML programming, the X Mon. is useful for entering the occasional ML routine, disassembling it to make sure it's entered properly, and relocating programs if there's a memory space conflict.

The X Mon. uses a method of decoding commands similar to BASIC's jump table. Commands are checked to be certain they're valid, then converted to a number which is used to select an address from a table at $0960- 0995 (see fig. 1). These addresses are stored in the usual ML order: lo byte,hi byte. For example, 53 is stored at $0960 and 0B at $0961. This particular pair is the entry point for the "display memory" command (@). These bytes are stored on the stack, along with the contents of the status register. An RTI instruction causes the computer to jump to that address ($0B53), and the @ command is executed. When done, the computer goes back to the start of the X Mon. to get your next command.

If you have more than 4K of memory, you'll want to relocate the X Mon. to the top of your memory. The RELOCATE command will do most of the work for you. For example, if you have 8K of memory, type R1800=0800,0FFF. Now add $1000 to each address in the RTI table. Change $0B53 to $1B53, for example. There are a few other locations to change. $0996 and $0997 contain the address of the IRQ vector ($0BC7). This is where the computer goes when a BRK instruction is encountered. $0998 and $0999 contain $081E, which is where 'the computer goes after the BRK routine is completed. $0864 and $0865 contain $0809, the vector for an actual IRQ signal. These three addresses are modified in the same manner as the RTI table.

Light Customizing:

This section is for those who only want to make a few minor changes to the X Mon. First of all, any command letters can be interchanged by switching their addresses in the RTI table. Since the memory display command (@) is used so often, I switched its address ($0B53) with the address of the "U" command, which is not implemented ($084C). Be careful when doing this, because you're using the @ command to change its own vector. "U" is now the memory display command. Here are some addresses that specify keys for various subcommands:

$0B74 (5E)-Display previous memory location (shift /N)
$0B78 (22)-Display memory as graphic char.(").
Also A,X,Y,P,K registers as graphic chars.
$0B67 (0D)-Key to exit memory display(return).
$0B70 (0A)-Display next memory location (line feed).
$0C2F (0A)-Disassemble next 24 lines (line feed).
$099D (17)-Number of lines to be disassembled.

By changing the contents of these addresses, you change the key that executes the subcommands. For example, change $0B70's contents from 0A to 20. Hitting the space bar will now print the contents of the next memory location.

Medium Mods:

Medium mods are more powerful than light mods, but may require some trade-offs. Make these changes:

$0B81 (60) to 8B
$0B88 (4C) to EA
$0B89 (60) to EA
$0B8A (0B) to EA

Note: you must make these changes with the ROM monitor.

Now the "@" command will automatically print the next address when you change the contents of an address or use (") to see the graphic equivalent of an address. You can type in ML programs without hitting the LINE FEED key after every byte. You can search for an ASCII string by holding down the (") key. If you change the A,X,Y,P, or K registers, an extra address will be displayed, but that is a small drawback, in my opinion.

Using a combination of changing $099D (17), which controls the number of lines disassembled and changing:

$0D1D (20) to EA $0D28 (20) to EA
$0D1E (07) to EA $0D29 (07) to EA
$0D1F (0B) to EA $0D2A (0B) to EA

You can eliminate two blank lines in the disassembly display and increase the number of lines displayed. The trade-off in the case is that you must exit the disassembly with a "return" before doing any other command. If you don't want to give up this convenience, you can still eliminate one blank line by changing only $0D1D-0D1F.

Here's a mod that intercepts data being loaded in the checksum format and stores it in a section of memory you specify. This could be useful for those programs located in areas of memory where you don't have RAM. Programs loaded this way will probably not run because they've not been relocated, but at least you can look at them with the disassembler. Also, the data is loaded into consecutive memory locations. If the program being loaded is not in consecutive locations, some of the data will be out of place.

First, change:

$0F6A (91) to 20
$0F6B (DC) to 22
$0F6C (C8) to 02

This causes the LOAD routine to go to a subroutine at $0222, which you enter:

```
$0222 C8       INY
 0223 8D 30 02 STA $0230
 0226 EE 24 02 INC $0224
 0229 D0 03    BNE $022E
 022B EE 25 02 INC $0225
 022E 60       RTS
```

The start of the program loaded will be at $0230. Change the address in the second line ($0230) to change the beginning address of the load. After every load this address must be reset to its original value.

For those people who have modified their C1P video display and use a patch to the video routine, the X Mon. output vector is at loc. $0862-0863. Change this address ($FFEE) to the start of your video patch.

Since the X Mon. uses some of the same zero-page locations that BASIC uses, the two can't be easily used together. A program can be written to restore locations $D0-D7 when going from the X Mon. to BASIC, but I prefer the more elegant solution. You can change forty locations in the X Mon. (see fig. 2) and change its zero-page useage from $D0-D7 to $50-57. Simply use the X Mon. to display the addresses listed in fig. 2. Change $D0 to $50, $D1 to $51, etc. $50-57 are locations in the input buffer. They're only used when you're entering BASIC lines (I think).

One more thing: My tape of the X Mon. had a bug at loc. $0FFE. $0FFD-0FFF should contain 4C 32 08 or the "Z" command won't work properly.

Heavy Mods:

Sorry, I don't have any yet. I have several ideas, such as a TRACE routine, a HEX-DEC converter, and burning part or all of the X Mon. into a PROM, but I decided to let you know what I've learned so far instead of waiting another two months. The following is a list of X Mon. subroutines. Good luck, and write if you think up some neat mods (or neat questions).

$0853 Input a char. If it's >20, or a CR, output to TV.
$0A78 Enter with number in A(hi byte) & X(lo). Prints 4-digit no.
$0AA3 Convert 2-digit hex input to binary in $E7 & A.
$0AAC Convert contents of A to hex; print on TV.
$0AC2 Shift lo nybble of $E7 to hi nybble; put lo nybble of A in lo nybble of $E7 put $E7 in A.
$0ADA Get 1 char; if it's <1 or>8, call $0AFF sub. If it's ASCII for 1-8, convert to binary 1-8 (for breakpoints).
$0AE7 Input a digit 0-F. Prints digit, converts ASCII to binary, stores no. in A reg. If input isn't 0-F, print char. and call$0AFF sub.
$0AFF Illegal command: Print "?" and go to start of X Mon.($0809) Skips initialization of break points.
$0B07 Output CR, LF.
$0B11 Store 4-digit hex input (XXXX) in $DA, $DB.

$0B1C Store input (XXXX,YYYY) in $DC-DF.

$0B36 Increment address in $DC,$DD. If $DC,DD > $DE,DF, then go to start of X Mon. ($0809). Otherwise, RTS.

$0B48 Store input XXXX in $DA,DB; printXXXX. Store input YYYY in $DC,DD; print YYYY. Store input ZZZZ in $DE,DF; print ZZZZ.

$0C51 Output a space.

$0E29 Prints contents of $D8,D9 as hex address.

$0E47 Print Y reg, X reg. as a 4-digit no. Enter at $0E48 to print A reg., X reg.

$0E70 Divide: Number in $DC,DD by $DE,DF. Remainder in $D8,D9.

$0E88 Multiply: $DC,DD times $DE,DF. Overflow in $D8,D9.

$0EAB Subtract: $DC,DD minus $DE,DF. Results in X(lo),Y(hi).

$0EB7 Add: $DC,DD plus $DE,DF. Results in X(lo), Y(hi).

$0F7F output "ERR" message.

$0F98 Add A reg. to number in $DA,DB. Put contents of $E7 in A reg.

$0FA7 Input 1 byte. If LOAD flag is on, output to TV. If LOAD flag is off, go to start of X Mon.($0809).

Miscellaneous Information:

Bootstrap loader	$0700-07EF
Main program	$0800-0FFF
Data table	$0866-099B

$0855	JSR FFEB	Input
$0861	JMP FFEE	Output
$0EC3	JSR FFF7	Save
$0F3B	JSR FFF4	Load
$0F43	JSR FFF4	Load
$0FA7	JSR FFEB	Input

Zero-Page Useage:

$D0-D7 Used in "Q", "N", "W" routines.

$D8 Used in "Q", Index for break table.

$D9 Breakpoint number(1-8).

$DA,DF Addresses for commands (XXXX, YYYY, ZZZZ)

$E0 A register stored here after BRK.

$E1 X ' '.

$E2 Y ' '.

$E3 P ' '.

$E4 K ' '.

$E5-E6 Location of last break.

$E7 Misc. uses.

$E8-EF Bytes displaced by insertion of BRK (00) command.

$F0-FF Break table.

fig. 1
X MON. RTI TABLE
($0960-0995)

@-	$0B53
A-	$0BB3
B-	$0C9A
C-	$0CBF
D-	$0CD2
E-	$0C57
F-	$0DA3
G-	$0BC1
H-	$0E33
I-	$0C12
J-	$084C
K-	$0BAF
L-	$0F43
M-	$0D91
N-	$0D35
O-	$0E29
P-	$0BB0
Q-	$0D14
R-	$0DB7
S-	$0EC3
T-	$0C6E
U-	$084C
V-	$0F3B
W-	$0D7E
X-	$0BB2
Y-	$0BB1
Z-	$0FB7

fig. 2
BASIC COMPATIBILITY CHANGES.
($D0-D7 to $50-57)

$099F	$0A95
$09A7	$0A98
$09A9	$0A9E
$09AB	$0D18
$09B4	$0D1C
$09DC	$0D3E
$09E0	$0D61
$0A01	$0D8E
$0A06	$0DBD
$0A1C	$0DC1
$0A21	$0DCC
$0A27	$0DD0
$0A29	$0DDE
$0A40	$0DED
$0A44	$0DF2
$0A4C	$0DF7
$0A50	$0DFD
$0A5A	$0E19
$0A83	$0E1D
$0A85	$0E21

015

*****CORRECTIONS*****

We have been informed by Sidney Sosin that in his letter, July, 1981 issue of Peek (65), page 13, contained a typographical error. The references to addresses $5025 to $5030, immediately following the listing of code changes, should have been $5425 to $5430. The address 5425 in the change listing is correct.

Peek (65)

* * * * *

LETTERS

ED:

After reading your June issue and Mr. Schaffer's letter, I decided to write (my first). The Utilities (MPUTIL) Mr. Schaffer wrote about, are put out by Ohio Scientific and come with an instruction manual on its use. The utilities occupy a single track (track 0) and the minimum hardware required is a serial based computer with 16K RAM at partition one and an additional 4K RAM at hex E000. There are three utilities:

1. System configuration verification, reporting all 16K RAM blocks on line in all operating partitions.

2. Two types of memory tests (random and bit), usable in any partition.

3. A system monitor capable of multiple partition operation.

These utilities are the only ones I know of that allow you to test all the partitions from another partition. The only time I have had any problem is when I was not paying attention to what I was doing and tried to test memory (random test) from $200 to $FFFE, the disk went crazy and wiped the disk, the printer started to print control characters then the system hung. After turning off the printer I tried to reboot the system but (believe it or not) I could not even get the reboot prompt (H/D/M) I turned the computer off, waited a few minutes then on and every thing was ok. Do you have any idea what happened?

While I am writing, I would like to ask you a few questions. 1. Do you know of any way to access another partition under 65U? 2. Is there any way to run 65U on one partition and 65D on another?

I have to tell you how much I like your articles on how to use the assembler. I have read about fixes for Level-3 but have been unable to find anyone who knows about them. 3. Can you tell me how I go about getting the fixes? I have found software for my system (C-3B w/dual floppies) to be scarce to say the least, so have started to keep a list of software and documentation, where to get it and a description of its purpose, I would be happy to share what I have found if anyone is interested.

For anyone looking for a good resident cross reference utility, I would like to recommend Software Consultants REF. After using this program for a couple of months, I have found it to be very good with one exception, there is a bug in it (at least my copy). The bug shows up in lines like this:

```
10 INPUT %I,IL
```

The channel variable shows up correctly, but the IL variable shows up without any line number. I wrote Software Consultants about a month ago but so far I have received no reply.

Mike Fowler
Fontana, CA

Mike:

First the questions: 1. I don't , but I am sure some readers do- how about it, readers?

2. Obviously, when you run LOAD 48, you get a limited version of 65D up and running, but I would be afraid to do much with it. Who has gone further? I hear 65D hard disk drivers will be out this year, that will help.

3. Watch this space for more info on fixes for level-3. Jim Sanders and others are at work.

I am sure you will hear from Software Consultants soon. They are very busy due to trememdous success of their 65D disassembly, etc. and the boss becoming a new father, but, they are a quality outfit. Please do not make this your last letter!

AL

* * * * *

ED:

It occurred to me the same day I received the June issue that even though I had made some changes to OS65D to run programs with 14 character names, I would be limited to only 32 files per 5 1/4 diskette. Then I read the article by "Stretch" Manley. I immediately fixed all my just discovered fears. I had to do some detective work (it seems the world of OSI is full of that kind of adventure) to figure where all the locations were in the 5 1/4 format.

Locations $2DE1, $2DE3, $2DF1, $2DE4 and $2E07 were all on track 1. Location $21A8 was on track 5 and $22AB was on track 6. The last two are not in the OS but are parts of BASIC. Mr. Manley's help was welcome and at just the right time.

One thing he said, though, puzzled me. His statements -- "CAUTION!!! Don't change the name of 'BEXEC*' in any way."--just aren't true. I have been following a routine published in a back issue of AARDVARK Journal to change the name to "E". The routine CALLs track 1 into memory and changes the portion which clearly shows "RUN"BEXEC*" on track 1 when EXAMined to "RUN" (here you put up to a 6 character file name). The name is limited to only six characters since that was the original space provided for the first program to execute. The reason for choosing "E" as the name is, as was stated in AARDVARK Journal,--it's easy to type.

I'd better mention my system since this is the one thing that really bugs me about reading any periodical on computers--48K C4P DF.

Here's something for anyone to think about who uses 65D. There's a bug (described in the AARDVARK Commented Listing) in the PUT/GET BASIC overlay (5 1/4 track 12,4) for random file handling: "The random access calculation routine would access the wrong track if a record was accessed which appeared in a track greater than 15 in a large file. This error occurred because a binary increment was used on the decimal track number." It's nice to know what I have already discovered. I call it doubling back, because that's what it does. I have uses for files, maybe even 38 tracks long! (I use a track 39 directory, you see.) But how in the hell can I make use of a LARGE random file if the routines have bugs? Does anyone have a suggestion? I have tried to work on it but my knowledge of dissembled code is rudimentary at best.

AARDVARK Journal recently ran one fix for a time eating inconvenience in the PUT/GET routine. It allows a read of any record on a given track of a random file WITHOUT the disk having to access the track for each DISK GET command. This has speeded up my file reading and writing programs greatly.

Here's another interest. The Digital to Analog Converter and associated software. The only programs I have that use this interesting gadget (that's all it is so far) on the C4P are from OSI. That disk is OK if you like kazoo music. (Actually , I learned the tenor part of quite a difficult Alleluja from a Bach Motet by inputing the four -part continuo with this program.) But again, can you really find joy singing with a kazoo continuo!? The real problem, inquiries about which have been ignored by OSI-BM (that's before MACOM), is in the program that plays the previously inputed tune. It asks if you want to designate the wave form. Well, come on folks, when you look at the program, it's easy to see that the POKEs made by your choices for voices are overwritten in the next line of the program. The program calls a wave shape table, at least I think that's what it is, into memory. And when I change the table, it does make a difference. But I still get nothing but square waves variously modified to the point of raucous distortion. Can the DAC produce anything but square waves as I am assured it can by the program itself and other literature?

The scroll POKE (9761) and the "Print At" routine by Phil Lindquist and Al Casper, respectively, have been my favorite articles (letters) so far. Both have helped me to make several slick graphics and text combinations in my programs. My favorite is a histogram program which builds the graphs on the screen from data input from disk. If anyone would like to share programs, an original author software exchange might be initiated.

Ross C. Votaw
Springfield, OH

* * * * *

ED:

I would like to comment on the D=D+7*(D>9) statement in my recent letter, June 1981, and some other things. First, the values of the part in parenthesis are 0 for false and -1 (not +1) for true. This is because 0 is reasonable for false and true is NOT false of the two's complement of 0 which equals -1. Second, I am not the originator of this kind of statement. To see a very extensive use of the statement form look in MICRO (March 1981, p. 7) "A 6502 Assembler in BASIC" by Edward Carlson. Lines 221, 223, 287, 292, and 1992, all use variations of this form of statement. This program can be very informative just for figuring the uses of this logic even if you already have an assembler. I don't know where Ed got the idea from and I can't reach him because he's helping out a well known computer company in Ohio between terms at MSU right now.

Third, the program by Curtis Anderson (Peek (65) June 1981) sounds like a crude, early version of an article by Morris and Finkbeiner (Micro Nov. 1980). It is not nearly as refined or as neat as their program, otherwise it does the same thing. Fourth, I really do appreciate the articles that I have seen and I hope everyone keeps it up because OSI may add some changes soon (I have heard that they are now shipping out BASIC-IN-ROM on EPROM as they have used the amount of ROMs they contracted for) and our input might be significant enough to make OSI's system what we want it to be.

Dale Mayers
Lansing, MI

* * * * *

ED:

```
PRINT 12 AND 7
PRINT 12 OR 7
PRINT NOT 12 OR 7
Z=A=B=C
```

are all valid BASIC statements and are explained in detail by Greg Yob in March 1980 Creative Computing on page 160.

Earl Morris
Midland, MI

* * * * *

ED:

Regarding Jack McKay's indirect file problem, in Vol. 2, No. 7. WP-2 does not seem to recognize the same termination character that OS65D recognizes. WP-2 uses 29 ($1D) while OS65D uses 93 ($5D). The procedure I have found is to put the BASIC program in RAM then poke in a 29 in RAM right after the OS65D termination character by entering the following BASIC line: POKE (PEEK(9106)*256+PEEK(9105)),29. The PEEKS get the address of the end of the just completed indirect file.

Tim Boege
Cedar Falls, IA

* * * * *

ED:

Your challenge in Column One of the June 1981 issue, to send you the best use of logic in parenthesis got my attention, even before I read Dale Mayers' interesting little Hex-Dec converter on page 7. In the line D=D+7*(D>9), the expression D>9 can only return a Boolean value of true or false. It is in parenthesis only to force its evaluation before the multiplication operation is performed. In OSI Microsoft BASIC, for some reason which I have never grasped, a true statement returns the value -1, not the value 1 as you state. In direct mode, do PRINT 10>9 and see what is returned. (Did Sanders really tell you it returned +1?) Now you know why BASIC programs which require a truth test to return 1 (and not -1) will not work on OSI systems. Try Mayers' program on Applesoft or Integer BASIC for Apple systems and you will see that it will not work.

In Mayers' program, the segment of line 9010 which is D=D+7*(D>9) has only one reason that I can see. It allows the test to appear on a multi-statement line. The expression will always have the value of 0 or -1 which permits the execution of the remaining statements on the line. When D>9 evaluates to 0, the statement is actually wasteful since D=D+(7*0), or D=D+0, involves a calculation which does nothing. If one were to substitute on the same line the synonymous (and clearer?) statement IF D>9 THEN D=D-7, while D<9 the remainder of the line will not be executed. If one has a minimum of memory such economizing acrobatics may be justified, but they do cause headaches when one tries to read the program a year later. My suggestion for the best use of a logic test of the sort illustrated is to put it out of mind. The next best use is to reserve it for generating a unary negative value by setting up a relation which will return a value of true (-1). In X=X+(X<Y), for example, X is decremented by 1 when the value of Y exceeds that of X because the addition of a unary negative results in a subtraction, i.e. X=8+(8<9) sets X to 7 by means of the operation 8+-1. The assignment X=X-(X<Y) increments X by 1 when the relation X<Y becomes true. These operations are, of course, valid for OSI Microsoft BASIC because of the unusual value of -1 for truth.

(By the way, if you have an uncorrected BASIC on your WP-2 disk (mine is 1 May 1979 Release), it will not perform addition on a unary negative. PRINT 6+-6 will return 12 instead of 0, and PRINT 6--6 will return 0 instead of 12. The long pause in execution indicates a bug somewhere. I wrote OSI for a fix, and, yes, they did answer. Their two-line reply: "BASIC is not supported in WP-2. Will work ok if unary minus not used, e.g. 6-6." Brilliant fix! I only solved the problem by putting a good 65-D BASIC over the bad WP-2 version.) To rewrite Mayers' program, put in a line 9005 IF D>9 THEN D=D-7, delete amusing D=D+7*(D9) from line 9010, and you have a good Hex converter which will run on more than OSI Microsoft BASIC. Are there other BASICs with the -1 quirk?

In the October 15th, 1980 issue, a letter from Kurt Gritter of Milwaukee offered a fix that would enable the use of port #5 instead of #3 for the serial 430 board. It will not work on my 430B board. Can he or anyone tell me why? The line of BASIC that he wants to put in BEXEC* contains the statement PEEK(L). Is this a possible BASIC statement? If an error, how should the line read? I have just purchased a new serial board from D&N which is wired for DEV #8. I can output to #8 on my 65-D but not on WP-2; the PRINT #8 command outputs nothing and causes the cursor to hang. Does anyone have a fix for this?

Can anyone out there tell me if it is possible to output a formatted WP-2 file, of longer length than can fit into an Indirect File, onto a 65-D data file? I can do it through the Indirect File by chopping an 11 track WP-2 text into small chunks, but this is very time consuming. I would like to do it in one operation. Also, has anyone tackled the problem of getting WP-2, 65-D and 65-U files into the UCSD Pascal Editor?

Let me put in a plug for the SANDERS MACHINE LANGUAGE SORT/MERGE which I bought from you last fall. Thanks to its efficiency, I completed a long dictionary project without a hitch, and I have just completed concordances for two medieval texts. Anyone with long lists to sort will do himself a favor by getting this "goodie".

M. Roy Harris
Charlottesville, VA

Roy:

Seems to me the Hex-Dec convertor with (D>9) runs faster than my BASIC effort with IF-THEN construct, but maybe that's cause of general looseness of much of my output. Anyway, even if it's only "amusing", what's wrong with being amused?

AL

* * * * *

ED:

This is in answer to Tom Westhoff's letter from the June issue of Peek (65), pages 10 and 11.

Tom's letter raised a storm of interest here and I have been selected to open communications. There are four highly modified C1-P and one modified C4-P in our group, and we have been discussing just such a monitor ROM as discussed in Tom's letter. One local C1-P now has a modified Aardvark C1S, two have unmodified C1S, and one has the original OSI monitor. I have a CEGMON (Aardvark C2E) in my C4-P, and have been loading and playing with the monitor routines in Ken Skier's byte book "Beyond Games". With that variety of monitor routines we were forced to the position you describe, and will be pleased to contribute what we can to Tom's effort.

To Comment on Each Item:

1. The keyboard routine changes effected in either of the ROM monitors from Aardvark are proof that this item should be very high on your priority list.

2. Good idea, but would prefer a default 32 to permit a one-key screen clear....(CTRL rubout maybe??).

3. This one doesn't rate much priority with this group since I am the only one with color.

4. The rubout backspace in the Aardvark monitors is a genuine pleasure. This feature deserves priority.

5. Isn't it almost as convenient to retype the line number?

6. A definite must! The CEGMON monitor provides this feature for the C4-P, and it adds considerable capability via "hooks" to the vectored routines.

7. I would hate to see monitor ROM space devoted to this problem. A replacement BASIC ROM is available

8. Can be implemented without too much code, but would rate a low priority.

9. I've grown to love the editor in the CEGMON, but I'm sure it is one among many suitable editors.

10. Agreed.

11. No comment here. We don't have any experience or background .

12. Same as #11.

13. Desirable, and shouldn't be difficult to implement.

14. This one requires more thought!

The monitor display formatting and functions described in Skier's book should be carefully studied. Some really fine stuff here, keep us posted on inputs and progress, and if we can help in a specific way, call on us. We have disassembled all the monitors mentioned, and we can evaluate candidate ROM routines on several different OSI configurations. For the largest market, you might consider a C1P keyboard routine.

Warm regards and the best of luck.

S.C. Dodd
Alamogordo, NM

* * * * *

ED:

During a pause from several hours of working with my computer, I left my office to chat with my wife for a few minutes. When I returned to my programming I stopped to reflect on the amount of time I spend with my computer, and how my wife never complains. A poem came to me which I think expresses the lot of many of your readers, and I would like to share it with them.

I dedicate this to all the understanding wives out there.

UNDERSTANDING WIFE

There's no computer like my own,
It's really quite terrific,
It's made by folks in O-hi-o,
Ohio Scientific.

With switches thrown and power on,
I spend my idle time,
Six hours a night I'm locked away,
Debugging line by line.

A can of Raid is at my side,
When switches are turned on,
While tracking down those nasty bugs,
I'm often up past dawn.

From TV tube to CRT,
I've reached a change in life,
From football games to program code,
A widow is my wife.

My wife is understanding though,
She's conscious of my need,
For printers, listings, program code,
And high speed paper feed.

I've had my system two years now,
My wife is with me still,
She doesn't seem to mind too much,
I hope she never will.

C.Emerson-Henry

* * * * *

******RUMOR******

A major OSI software vendor will soon produce a routine allowing OSI systems to read standard IBM format disks under CP/M, opening wide the doors to all the software you read about in the magazines.

*****CORRECTIONS*****

Another error has been brought to our attention by Pete Hitt, in reference to his letter to the Editor, in the August 1981 issue, pages 15 and 16. Pete tells us that the first lines in paragraphs 3 and 4, both contain reference to a 2713 ROM. This designation should be 2316.

Peek (65)
* * * * *

NEW PRODUCTS
*****ANNOUNCEMENT*****

Problem-Solving Principles For Programmers: Applied Logic, Psychology and Grit, is written to improve computer programmers' problem solving techniques. There are three other books specializing in solutions unique to problems encountered by FORTRAN, PASCAL and BASIC programmers.

Written by William E. Lewis and published by the Hayden Book Company, Inc. of Rochelle Park, New Jersey, each book contains five different chapters, similar only in the consistency of three interwoven conceptual threads: general problem solving, program problem solving and psychological influences on the overall problem-solving process.

PROBLEM-SOLVING PRINCIPLES FOR PROGRAMMERS: APPLIED LOGIC, PSYCHOLOGY AND GRIT by William E. Lewis; 1980; 6 x 9; paper; 163 pp.; ISBN-8104-5138-7; $9.95. Also available in PASCAL, 5767-9; FORTRAN, 5430-0 and BASIC, 5200-6.

AD$

65D Magic Boot: Adds renumber, full cursor editing, memory pack, screen clear and color background to BASIC. All routines are superfast machine code and handle All cases. These are not USR (x) routines, but extensions to BASIC itself. All you do is boot and these commands are ready to use--in immediate mode or in your programs. Also includes new BEXEC* and disk utilities. $50.00. UNIVERSAL SYSTEMS, 2020 West County Rd. B, MPIS., MN 55113.

65D Segs: Adds segmentation commands to BASIC. Allows segment calls (like GOSUB'S) to subroutines stored on disk. By nesting calls, LARGE programs may be written and will run in (24K) memory. (Requires Magic Boot) $25.00. UNIVERSAL SYSTEMS, 2020 West Country Rd. B, MPIS., MN 55113.

FOR SALE - Challenger C1P, cassette, video, & printer interfaces, 8KRAM, cassette blanks + documentation $350. Call Paul- W:(202) 659-0590 H:(703) 437-8574

Z-FORTH IN ROM by Tom Zimmer
5 to 10 times faster than Basic. Once you use it, you'll
never go back to BASIC! $ 75.00
source listing add $ 20.00

OSI FIG-FORTH True fig FORTH model for 0S65D with fig editor
named files, string package & much more $ 45.00

TINY PASCAL Operates in fig-FORTH, an exceptional value
when purchased with forth.
TINY PASCAL & documentation $ 45.00
FORTH & TINY PASCAL $ 65.00

SPACE INVADERS 100% machine code for all systems with
64 chr. video. Full color & sound on C2, 4P & 8P systems. The
fastest arcade program available. $ 14.95

PROGRAMMABLE CHARACTER GENERATOR $ 99.95
Use OSI's graphics or make a complete set of your own! Easy
to use, comes assembled & tested.
2 Mhz. boards $109.95

PROGRAMMABLE SOUND BOARD $ 74.95
Complete sound system featuring the AY-3-8910 sound chip.
Bare boards available. $ 29.95

32/64 CHARACTER VIDEO MODIFICATION $ 39.95
Oldest and most popular video mod. True 32 chr. C1P, or 32/64
chr. C4P video display. Also adds many other options.

ROMS!!!
Augment Video Mod with our Roms. Full screen editing, print
at,selectable scroll, disk support and many more features.
Basic 4 & Monitor $ 49.95
Basic 3 $ 18.95
All 3 for $ 65.00

65D DISASSEMBLY MANUAL. by Software Consultants
First class throughout. A must for any 65D user.
$ 24.95

NUMEROUS BASIC PROGRAMS, UTILITY PROGRAMS AND GAMES ALONG WITH HARDWARE PROJECTS. ALL PRICES ARE U S FUNDS. Send for our $1.50 catalogue with free program (hardcopy) Memory Map and Auto Load Routine.

CHARGEX VISA master charge

OSI Software & Hardware
3336 Avondale Court
Windsor, Ontario, Canada N9E 1X6
(519) 969-2500

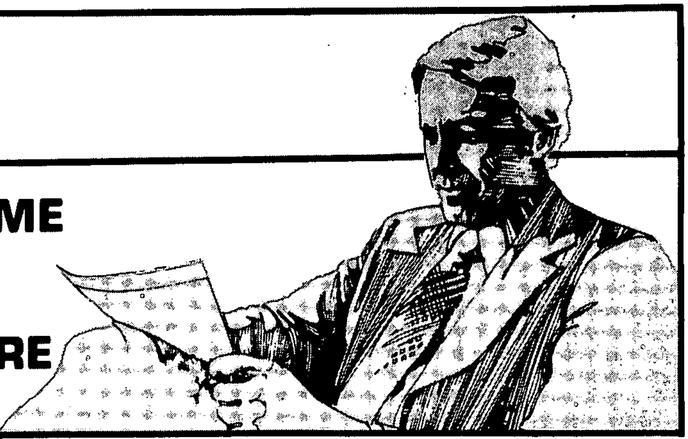
3281 Countryside Circle
Pontiac Township, Michigan 48057
(313) 373-0468

progressive computing

SAVE

YOURSELF SOME TIME AND TROUBLE

WITH OUR 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, so be sure to keep us in mind for all of your upcoming projects.

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 dissassembly listings, complete and clear comments on most every line, 10 pgs. of computer generated cross reference listings, and more! Praised by many OSIers who couldn't believe it 'til they bought one. A deal at $25.95.

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 5¼" or 8" floppies. This one will save your sanity, and cut out hours of wasted time. Yours for $31.95.

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 $89.95.

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 $25.95, with extensions, $29.95.

6. **GRAPHICS RESOLUTION BOOSTER**
   An ingenous 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 $12, 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.

SOFTWARE CONSULTANTS
HARDWORKING *software*

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) @ $25.95 ea.
_____CROSS REFERENCE UTILITY(S) @ $31.95 ea.
Disk size 5¼"_____or 8"_____
OS-65D_____or OS-65U_____
_____SPOOLER/DESPOOLER UTILITY(S) @ $69.95.
Std. parallel interface_____
Serial interface w/430 board or equivalent_____
Serial interface w/CA10X board_____
_____FIG FORTH(S) @ $89.95 ea.

_____VIDEO ROUTINE(S) @ $25.95 ea.
With software extensions @ $29.95 ea.
_____SOURCE CODE @ $12.00 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

DELIVER TO: