PEEK (65)

The Unofficial OSI Users Journal

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

$1.75

NOVEMBER 1982
Vol. 3, No. 11

INSIDE:

C1P CORNER	2
HYBRID DISK DIR. SORT	5
65U V1.42	9
CEGMON	20

Column One

A recently popular book starts with the words "Life is difficult." Seems obvious, but when the difficulty of life strikes close to home, we are still surprised.

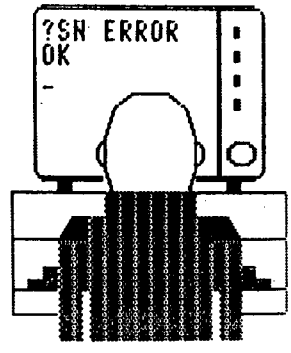
What brings all this up is the news that both Digital Technology and H/B Computers, old friends and co-workers in the OSI community, have gone out of business. Though not yet confirmed to us in writing, it seems pretty definite from here.

On a brighter note, Bob Ankeney of Generic Computer Products has written us a letter pledging "a high level of support for OSI users, with an extensive line of quality hardware and software products" for the hobbyist market. Ankeney promises a one-year warranty and exciting new product announcements by the year's end.

Meanwhile, M/A-COM OSI has come out with OS-65U V 1.43, significantly improved over 1.42 with:

Masterkey 2301 system;
terminal type;
disk to multiple floppies;
SYSDIR for multiple-partition hard disks.

It seems, then, that M/A COM OSI will concentrate on upgrading the upper end of the hardware and software, while outfits like Generic, Aardvark and many others will continue to support the hobbyist.



To The Midnight Hackers

see page 16

PEEK(65), of course, will support both. This issue is a good example. We have lots of hard information for the hobbyist user, in our letters and the C1P corner, plus a description of the CEGMON monitor by its originators.

A 65D directory sort program is presented and thoroughly explained -- if you use 65D, have a look at this article, and think how you can expand on the basic concepts it outlines... then write to us with your best ideas!

A user with a 230E multiuser system running 65U V 1.42 explains some important enhancements in what seems to have been a rather smooth installation.

Finally, NBTel of Canada describes a truly sophisticated system running PASCAL networking on a C-3B.

About the only system we haven't described or touched upon in this issue is YOURS. Why haven't you written to us describing what you use your computer for? We really would be interested. The future of M/A COM OSI will be determined largely by the community of users and what we do and develope. All the factory can do is provide us with the tools. We must do the work.

ae

C1P CORNER

By: David A. Jones
8902 SW. 17th Terrace
Miami, FL 33165

Assembler/Editor

Some things I've found out but don't remember seeing elsewhere are a couple of memory locations of interest. The Assembler/Editor input buffer length limit is held in location $022B for disk versions or $1382 for cassette and is normally $38 (decimal 56) presumably to keep the video display from wrapping around on 72 character displays when assembled programs are listed. If you have an 80 column printer you may wish to change this location to $40 and gain an additional 8 character spaces for remarks.

CA 0200=07,1, change $022B to $40 and SA 07,1=0200/8.

Extended Monitor

In the September '81 issue of PEEK (65), Kerry Lourash outlined some enhancements for the Extended Monitor. To go one step further, change locations $0B73-0B82 from:

```
C9 5E F0 1D C9 22 D0 08 B1 DA
20 61 08 4C 60 0B
```

to:

```
C9 2D F0 1D C9 20 D0 08 B1 DA
20 61 08 4C 8B 0B.
```

Using this change will now automatically display the ASCII equivalent of the memory location opened with the shift P command (I use U vs shift P) when the space bar is depressed and then increments the address to the next location. No shift key or LF required now. To back up, depress the - (minus sign) rather than shift N.

PEEK (65) is published monthly by DBMS, Inc.
Owings Mills, MD 21117.
Editor - Al Peabody
Technical Editor - Dickinson H. McGuire
Asst. Technical Editor - Brian Hartson
Circulation & Advertising Mgr. - Karin Q. Gieske
Production Dept. - A. Fusselbaugh, Ginny Mays
Subscription Rates

US (surface)	$15
Canada & Mexico (1st class)	$23
So. & Cen. America (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

For disks, the locations are $1AC5-1AD4 and the new code is C9 2D F0 22 C9 20 D0 08 B1 CA 20 A0 17 4C DD 1A. Make the changes and then SA 10,1=17 00/8.

Keyboard

In the September '82 issue of PEEK, Carl King requested help in locating a new keyboard routine that he could use with OS65D3.2. I don't think Brian properly understood the problem Carl is faced with (or maybe I don't). OS65D3.2 and below, with polled keyboards, depend on a routine in the monitor ROM to scan the keyboard for depressed keys. This routine works rather strangely when the shift lock key is not depressed in that the left shift and right shift keys generate different characters when held down in conjunction with another key. This enables OSI systems to utilize the entire 128 character ASCII set with only 53 hardware keys. The drawback is that the typist must remember which shift key to use to get the desired result.

James Loos published a new keyboard scan routine as part of a terminal program in the April '81 issue of PEEK (65). I extracted the portion of interest to me (I already had a terminal routine) and put it in EPROM and still use it today with 65D. Unfortunately, James didn't have an assembler so only provided a disassembled listing. The following is the reworked source code for the part that I adapted to my C1P. This can be assembled to the top of memory, saved on a system disk, track 6 sectors 2 and 3 are convenient on a 5-1/4" system, and called automatically by BEXEC* as follows. Carl mentioned that he had a 48K system so the example uses the last 1K block. This may be changed according to the requirements of the user. Add to BEXEC* '22 DISK!"CA 2500=06,2":DISK!"CA BF00=06,3'. Track 6, sector 2 was saved from $2500 for 1 page to preserve the new keyboard input vector at $2531 which used to be JSR $FD00. Remember that shift p will now give an upper case p when in the lower case mode so the use of a different delete character is recommended. I use rubout.

Reserved Space

When a disk is booted, the memory size routine starts at the highest possible RAM location and checks downward until read/write memory is found. This beginning value is nominally $BFxx and is stored in location $2277. Changing $BF to $77 will automatically reserve 2K of memory at the top on a 32K system for your own use without resorting to the 'Change' routine or POKING 133 etc. This is on track zero so must be changed with the TRK 0 utility. See Stretch Manley's article in September '81 if you're not proficient in this area. Copy track 0 to $4200, change $4277 to $77 and then save it back again. My screen editor is located at $7800 so all of my development disks have this feature and the editor is called by BEXEC* upon boot. Carl might use $BE as the above keyboard routine requires only one page of his 48K.

Directory

Sending the output of the 'DIR' utility to a line printer is a nice idea if you like to keep track of what's on each disk, but unfortunately, OSI didn't provide us with a method to identify each disk. I modified 'DIR' to prompt me for the disk ID when a listing was printed but soon got tired of having to answer the prompt. Examining the inner workings of the directory operation we find each entry in the directory is composed of 8 bytes. The first 6 are the name of the file and the last 2 are the numbers of the first and last tracks of the file represented in packed (4 bits/digit) BCD. Once an item is written into the directory no further check is made to see if the track numbers or file names are duplicated or legal. With this thought in mind we can use the first 8 bytes to store the disk ID as a permanent part of each directory. To do this, all of the present entries must first be moved back 8 bytes. This is easy to do with the routine that follows. The first entry in the directory becomes NR.xxx 0 - 0, where xxx represents the disk number. Since we all know OS65D is on track zero, we can ignore the duplication of the track number.

As long as we're playing around with the contents of the directory, why not go all the way and include an entry for the assembler/editor, extended monitor, and copy utility on track 13. This is most easily done with the EM by !CA 4000=12,1, changing the contents with the memory modification commands and then

saving it back to track 12. The ASCII display mod above is particularly handy for this now. Disks that are used only for applications or BASIC development can dispense with the ASM and EM thus freeing 5 tracks for program storage. Knowing how to create and/or modify a directory manually can be a godsend if you have ever accidentally written over track 12 or wished to expand an existing file without having to go through the delete/create rigamarole.

Copy

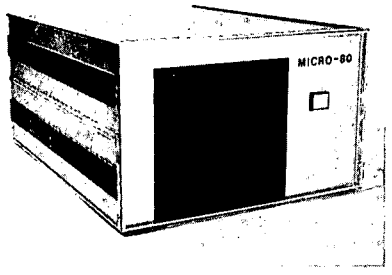
The copy utility by Jeff Dripps that appeared in the June and July '82 issues of PEEK (65), is vastly superior to OSI's copy utility. Since I never seem to have enough time to do all of the things I would like, I sent Jeff $10.00 and he sent me a disk containing the source and object code. Sure saved me a lot of time typing and searching for those typos. I had the head load mod already installed.

```
100 PRINT"DISK IDENTIFICATION,
    8-5-82
110 PRINT"MOVES DIRECTORY BACK
    ONE 8 BYTE BLOCK
120 PRINT"AND INSERTS DISK
    IDENTIFICATION IN THIS
    SPACE
130 DISK!"CALL 4008=12,1
140 DISK!"CALL 4108=12,2
150 PRINT:INPUT"DISK NUMBER";
    N$
160 IF LEN(N$)<3 THEN N$="0"
    +N$:GOTO 160
170 N$="NR."+N$
180 FOR X=1 TO 6:Y=ASC(MID$
    (N$,X,1))
190 POKE 16383+X,Y:NEXT
200 POKE 16383+7,0:POKE
    16383+8,0
210 DISK!"SAVE 12,1=4000/1
220 DISK!"SAVE 12,2=4100/1
```

```
    OS-65D VERSION 3.1.2
     -- DIRECTORY --

    FILE NAME     TRACK RANGE
    ----------------------------
    NR.012          0  -  0
    OS65D           0  -  1
    BASIC           2  -  6
    ASM             7  -  9
    EM+            10  - 11
    DIRECT         12  - 12
    COPY           13  - 13
    BEXEC*         14  - 14
    UTIL           15  - 15
    FUTIL          16  - 16
    SCRTCH         17  - 19
    ITCH           20  - 20
    BYTE           21  - 21
    KICK           22  - 22
    ETC            23  - 23
```

```
10        ;LOOS' KBRD,12-31-81
20        ;Revised 9-7-82 by
          D.A. Jones
30        ;
40        *=$2531
50        JSR KYBD
60        ; patched to branch
          new scan routine
70        *=$BF00
80        ;
90  KYBD  TXA
100       PHA
110       TYA
120       PHA
130       ;
140 KBRD  LDA #$01
150 CCCC  JSR $FCBE
160       JSR $FCC6
170       BNE EEEE
180 DDDD  ASL A
190       BNE CCCC
200       BEQ JJJJ
210 EEEE  LSR A
220       BCC FFFF
230       ROL A
240       CPX #$21
250       BNE DDDD
260       LDA #$1B
270       BNE GGGG
280 FFFF  JSR $FDC8
290       TYA
300       STA $0213
310       ASL A
320       ASL A
330       ASL A
340       SEC
350       SBC $0213
360       STA $0213
370       TXA
380       LSR A
390       JSR $FDC8
400       BNE JJJJ
410       CLC
420       TYA
430       ADC $0213
```

NEW FROM D & N MICRO PRODUCTS, INC.



MICRO-80 COMPUTER

Z80A CPU with 4MHz clock and CP/M 2.2 operating system. 64K of low power static RAM. Calendar real time clock. Centronics type parallel printer interface. Serial interface for terminal communications, dip switch baud rates of 150 to 9600. 4" cooling fan with air intake on back of computer and discharge through ventilation in the bottom. No holes on computer top or side for entry of foreign object. Two 8" single or double sided floppy disk drives. IBM single density 3740 format for 243K of storage on each drive. Using double density with 1K sectors 608K of storage is available on a single sided drive or 1.2 meg on a double sided drive. Satin finish extruded aluminum with vinyl woodgrain decorative finish. 8 slot backplane for expansion. 48 pin buss is compatible with most OSI boards. Uses all standard IBM format CP/M software.

Model 80-1200 $2995
2 8" single sided drives, 1.2 meg of storage

Model 80-2400 $3495
2 8" double sided drives, 2.4 meg of storage

Option 001 $ 95
Serial printer port, dip switch baud rate settings

Software available in IBM single density 8" format.

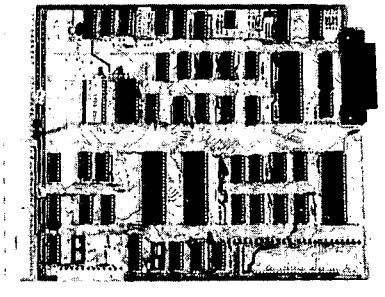
Microsoft		Digital Research		Micropro	
Basic-80	$289	PL/1-80	$459	Wordstar	$299
Basic Compiler	$329	Mac	$ 85	Mail-Merge	$109
Fortran-80	$410	Sid	$ 78	Spellstar	$175
Cobol-80	$574	Z-Sid	$ 95	SuperSort I	$195
Macro-80	$175	C Basic-2	$110	**Pascal**	
Edit-80	$105	Tex	$ 90	Pascal/MT +	$429
Mu Simp/Mu Math	$224	DeSpool	$ 50	Pascal Z	$349
Mu Lisp-80	$174	**Ashton-Tate**		Pascal M	$355
		dBase II	$595		

Convert almost any static memory OSI machine to CP/M® with the D & N-80 CPU Board.

Z80A CPU with 4MHz clock. 2716 EPROM with monitor and bootstrap loader. RS-232 serial interface for terminal communications or use as a serial printer interface in a VIDEO system. Disk controller is an Intel 8272 chip to provide single or double density disk format. 243K single density or 608K double density of disk storage on a single sided 8" drive. A double sided drive provides 1.2 meg of storage. DMA used with disk controller to unload CPU during block transfers from the disk drives. Optional Centronics type parallel printer port complete with 10 ft. cable. Optional Real Time Calendar Clock may be set or read using 'CALL' function in high level languages. Power requirements are only 5 volts at 1.4 amps. Available with WORDSTAR for serial terminal systems.

Includes CPM 2.2

D & N-80 serial $695
D & N-80 serial w/Wordstar $870
D & N-80 video $695
Option 001 $ 80
parallel printer and real time calendar clock



D & N-80 CPU BOARD

OTHER OSI COMPATIBLE HARDWARE

IO-CA10X Serial Printer Port $125
Compatible with OS-65U and OS-65D software

IO-CA9 Parallel Printer Port $175
Centronics standard parallel printer interface with 10 ft. flat cable

BP-580 8 Slot Backplane $ 47
Assembled 8 slot backplane for OSI 48 pin buss

24MEM-CM9 $380 **24MEM-CM9F $530**
16MEM-CM9 $300 **16MEM-CM9F $450**
8MEM-CM9 $210 **8MEM-CM9F $360**
BMEM-CM9F $ 50 **FL470 $180**
24K memory/floppy controller card supports up to 24K of 2114 memory chips and an OSI type floppy disk controller. Available fully assembled and tested with 8, 16, or 24K of memory, with floppy controller (F). Controller supports 2 drives. Needs separated clock and data inputs. Available Bare (BMEM-CM9F) or controller only (FL-470). Ideal way to upgrade cassette based system

C1P-EXP Expansion Interface $ 65
Expansion for C1P 600 or 610 board to the OSI 48 pin buss. Requires one slot in backplane. Use with BP-580 backplane

BIO-1600 Bare IO card $ 50
Supports 8K of memory, 2 16 bit parallel ports may be used as printer interfaces. 5 RS-232 serial ports, with manual and Molex connectors

DSK-SW Disk Switch $ 29
Extends life of drive and media. Shuts off minifloppy spindle motor when system is not accessing the drive. Complete KIT and manual

D & N Micro Products, Inc.
3684 N. Wells St.
Fort Wayne, Ind. 46808
(219) 485-6414

VISA MasterCard

TERMS $2.50 shipping, Foreign orders add 15%.
Indiana residents add 4% sales tax.

Disk Drives and Cables

8" Shugart SA801 single sided **$395**
8" Shugart SA851 double sided **$585**
FLC-6 6ft. cable from D & N or OSI controller to 8" disk drive **$ 69**
5 1/4" MPI B51 with cable, power supply and cabinet **$450**
FLC-5 1/4 8 ft. cable for connection to 5 1/4 drive and D & N or OSI controller, with data separator and disk switch **$ 75**

Okidata Microline Printers

ML 82A Dot Matrix Printer $534
120 CPS, 80/120 columns, 9.5" paper width, friction or pin feed

ML 83A Same as 82A except **$895**
16" paper width, 132/232 columns with tractor feed

ML 84 Same as 82A except 200 CPS, **$1152**
16" paper width, 132/232 columns, 2K buffer, dot addressable graphics, with tractor feed

```
440         TAY
450         LDA $FDCF,Y
460 GGGG    CMP $0215
470         BNE KKKK
480         DEC $0214
490         BEQ LLLL
500         ;
510         LDY #$05
520 HHHH    LDX #$C8
530 IIII    DEX
540         BNE IIII
550         DEY
560         BNE HHHH
570         BEQ KBRD
580         ;
590 JJJJ    LDA #0
600         STA $0216
610 KKKK    STA $0215
620         LDA #$04
630         STA $0214
640         BNE KBRD
650 LLLL    LDX #$96
660         CMP $0216
670         BNE MMMM
680         LDX #$14
690 MMMM    STX $0214
700         STA $0216
710         LDA #$01
720         JSR $FCBE
730         JSR $FCCF
740         LSR A
750         BCS QQQQ
760         BNE OOOO
770         LDY #0
780         LDA $0215
790         CMP #$D0
800         BEQ NNNN
810         LDA $0215
820         BMI SSSS
830         CMP #$40
840         BMI SSSS
850 NNNN    LDY #$20
860         BNE SSSS
870 OOOO    LDY #$F0
880         LDA $0215
890         CMP #$D0
900         BEQ PPPP
910         LDA $0215
920         BMI SSSS
930         LDY #0
940         CMP #$40
950         BPL SSSS
960         LDY #$10
970         CMP #$21
980         BCS SSSS
990 PPPP    LDY #0
1000        BEQ SSSS
1010 QQQQ   TAX
1020        AND #$03
1030        BEQ RRRR
1040        LDY #$10
1050        LDA $0215
1060        BPL TTTT
1070        LDY #$F0
1080        BNE TTTT
1090        ;
1100 RRRR   LDY #0
1110 SSSS   CPX #$20
1120        BNE TTTT
1130        LDY #$C0
1140 TTTT   LDA $0215
1150        AND #$7F
1160        CMP #$20
1170        BEQ UUUU
1180        STY $0213
1190        CLC
1200        ADC $0213
1210 UUUU   STA $0213
1220        PLA
1230        TAY
1240        PLA
1250        TAX
1260        LDA $0213
1270        RTS
1280        ;
```

★

A HYBRID DISK DIRECTORY SORT

By: Sidney Sosin
1107 Arbor Lane
Glenview, IL 60025

Here's a simple hybrid BASIC/machine language program which will sort your OSI disk directories 'on disk' almost instantly, using a potent combination of system commands and a machine language sort subroutine.

OSI's disk operating system (OS65D) has a directory utility containing a sort routine, but it's all in BASIC and you can grow old waiting for it to finish. Even then, the result is not on the disk, so next time you want the directory sorted you must wait again.

Listing 1 sets forth the BASIC program, which pokes in the machine code and accesses the system commands. The assembly language version of the sort routine is in listing 2.

Using the Program

Insert the disk with this program in the "A" drive and the disk with the directory to be sorted in the "B" drive. If you have only one drive, first load the program into memory, then insert the disk to be sorted and answer "A" to the question "In which drive is the directory to be sorted?". Next, select whether you want the directory sorted by track or name. The program then gives you a chance to change your mind by asking "Ready?". On an affirmative response, it sorts the directory and prints out on the screen what is already on disk. The display is in four-column format, which allows even the largest directories to be shown compactly.

How it Works

The BASIC program pokes the machine language sort into memory at hex 7000, uses system commands to call in the directory from track eight, sectors one and two, to memory at 7100 to 72FF, then goes to the sort routine via the USR command, puts the sorted directory back on the disk, and finally, prints out the sorted version. If you don't want the directory restored on disk in name order, then add the following as line 435: IF S=0 then 460.

There is nothing magic about the memory blocks used in the program; you can use any you desire, providing you make appropriate changes in lines 90, 310 through 340, 440 and 450 of the BASIC program and reassemble the machine language program accordingly. If you are not up to typing or assembling and desire a custom version, I would be willing to provide one on your 8 inch disk for $7.50 postpaid.

The Machine Language Sort

Although the machine language sort is included in BASIC as the data poked into memory, it is much easier to understand if the assembly version in Table 2 is explained. The routine is a bubble sort, which is slow in BASIC but almost instantaneous in this application, since there are a maximum of 64 entries in the directory. Each eight byte entry is compared with the next and switched if the first entry is higher than the following one. Indirect addressing is used and the starting Y register value is a passed parameter from the BASIC program stored in 00E7 (decimal 231). If it's a zero, the sort begins with the first character in each entry and continues, character by character, until a difference is found. If the parameter is a six, then the sort only looks at the binary coded decimal track numbers which are the seventh and eighth characters in each entry. (The parameter is a six because when you start with zero, six is the seventh character. Confusing but important to remember.) The assembly language program in Table 2 contains comments which should explain the details of its operation.

This program has been a great help in rearranging the chaos generally present on my disks, which generally start out organized but end up in a shambles. An additional help is another hybrid program I wrote which creates, lists and searches a library of all my disks. Now if I can only remember to update it!

LISTING 1 -- BASIC PROGRAM

```
10 REM DIRECTORY SORT
20 REM COPYRIGHT 1982 BY SIDNEY SOSIN
30 REM GLENVIEW, ILLINOIS 60025.  ALL RIGHTS RESERVED.
40 DEFFNA(X)=10*INT(X/16)+X-16*INT(X/16)
50 FORX=1TO25:PRINT:NEXT
60 :
70 : REM POKE IN MACHINE LANGUAGE SORT ROUTINE
80 X=28672: REM = HEX 7000
90 FORY=0TO108
100 READD
110 POKEX+Y,D
120 NEXT
130 :
140 PRINTTAB(17)"DIRECTORY SORT":PRINT
150 PRINT:PRINT"This program will load the standard OSI
    directory format"
160 PRINT"into memory, sort it by track number and put the
    sorted"
170 PRINT"directory back on the disk, all in a few seconds."
180 PRINT:INPUT"In which drive is the directory to be sorted
    (A/B)";D$
190 DISK!"SE "+D$
200 PRINT:INPUT"Sort by track (1) or name (2)";S
210 :
220 : REM PASS PARAMETER TO SORT ROUTINE
230 IFS=1THENPOKE231,6:GOTO250
240 POKE231,0:REM 231 = HEX 00E7
250 PRINT:PRINT"READY";:INPUTA$
260 IFLEFT$(A$,1)<>"Y"THEN180
270 :
280 : REM SET UP USR JUMP AT 7000
290 : REM   AND BRING IN DIRECTORY
300 : REM   AT 7100 TO 7300
310 AD=28928: REM = HEX 7100
320 POKE574,0:POKE575,112
330 DISK!"CA 7100=08,1"
340 DISK!"CA 7200=08,2"
350 : REM STEP THROUGH DIRECTORY
360 : REM  IF THERE IS NO ENTRY (HEX 23, DECIMAL 35)
370 : REM   THEN POKE IN 77 (119 DECIMAL) AS TRACK NO.
380 FORK=ADTOAD+504STEP8: REM EACH ENTRY TAKES 8 BYTES
390 IFPEEK(K)=35THENPOKEK+6,119:POKEK+7,119
400 NEXTK
410 :
420 :  REM NOW DO THE SORT
430 X=USR(X)
440 DISK!"SA 08,1=7100/1
450 DISK!"SA 08,2=7200/1"
460 PRINT:PRINT"Directory is sorted.  You are now in drive
    ";D$;"."
470 :
480 : REM  PRINT OUT DIRECTORY
490 PRINT:PRINTTAB(18)"SORTED DIRECTORY":PRINT
500 FORI=ADTOAD+496STEP8
510 IFPEEK(I)=35THEN630
520 C$="": REM C$ WILL BECOME ENTRY
530 FORJ=ITOI+5: REM BUILD C$ CHARACTER AT A TIME
540 C$=C$+CHR$(PEEK(J))
550 NEXTJ
560 IFLEN(C$)<6THENC$=C$+" ":GOTO560: REM PAD SPACES
570 D1=FNA(PEEK(I+6)):D2=FNA(PEEK(I+7)): REM CONVERT TO ASCII
580 C$=C$+" "+RIGHT$(STR$(D1+100),2)+"-"+RIGHT$(STR$(D2+100),2)
590 :
600 : REM PROVIDE FOR FOUR COLUMNS
610 PRINTC$,
620 SP=SP+1:IFSP=4THENPRINT:SP=0
630 NEXTI
640 PRINT
650 PRINT:INPUT"ANOTHER SORT";A$
660 IFLEFT$(A$,1)="Y"THENRUN
670 END
680 DATA216,169,0,133,224,133,225,133,229,169
690 DATA8,133,227,169,113,133,226,133,228,169
700 DATA115,133,230,164,231,208,8,177,227,240
710 DATA16,201,35,240,15,177,227,209,225,208
720 DATA4,200,76,27,112,176,3,32,81,112
730 DATA165,227,208,2,230,226,133,225,24,105
740 DATA8,133,227,176,5,240,3,76,23,112
750 DATA230,228,165,228,201,115,240,23,76,23
```

```
760 DATA112,160,7,230,224,177,225,145,229,177
770 DATA227,145,225,177,229,145,227,136,16,241
780 DATA96,165,224,240,3,76,1,112,96
```

```
10                ; TABLE 2 -- MACHINE LANGUAGE SORT ROUTINE
20                ;
30 7000           *=$7000     ; ASSEMBLE AT HEX 7000
40 00E0=          FLAG=$E0    ; COMPLETED SORT FLAG
50 00E1=          DIR=$E1     ; POINTER TO DIRECTORY ENTRY
60 00E3=          DIR2=$E3    ; POINTER TO NEXT ENTRY
70 00E5=          TEMP=$E5    ; TEMPORARY SORT STORAGE
80 00E7=          SRTFLG=$E7  ; PASSED PARAMETER FROM BASIC
90                            ; 0=SORT BY NAME  6=SORT BY TRACK
100               ;
110 7000 D8             CLD             ; INITIALIZE
120 7001 A900     BEGIN LDA #0
130 7003 85E0           STA FLAG
140 7005 85E1           STA DIR
150 7007 85E5           STA TEMP
160 7009 A908           LDA #8          ; SECOND POINTER 8 BYTES OVER
170 700B 85E3           STA DIR2
180 700D A971           LDA #$71        ; DIR START ADDRESS 7100
190 700F 85E2           STA DIR+1
200 7011 85E4           STA DIR2+1
210 7013 A973           LDA #$73        ; TEMPORARY STORAGE AT 7300
220 7015 85E6           STA TEMP+1
230 7017 A4E7     INIT  LDY SRTFLG      ; GET SORT TYPE
240 7019 D008           BNE ST0         ; IF ZERO SKIP FIRST TEST
250 701B B1E3     START LDA (DIR2),Y    ; GET 2ND TRACK NO.
260 701D F010           BEQ ST2         ; IF IT'S TRACK ZERO, SWITCH IT
270 701F C923           CMP #$23        ; IF BLANK ENTRY, SKIP IT
280 7021 F00F           BEQ NEXT
290 7023 B1E3     ST0   LDA (DIR2),Y    ; IF NAME SORT, START HERE
300 7025 D1E1           CMP (DIR),Y     ; COMPARE FIRST ENTRY
310 7027 D004           BNE ST1         ; IF NOT SAME, TRY NEXT TEST
320 7029 C8             INY             ; SAME - TRY NEXT CHAR
330 702A 4C1B70         JMP START
340 702D B003     ST1   BCS NEXT        ; 2ND ENTRY GREATER
```

OSI COMPATIBLE PRODUCTS

56K 2-MHz Ultra Low Power CMOS Static Memory Board **MEM-56K $850**

Partially Populated Boards (Specify address locations required) ... **MEM-48K $750**
MEM Board uses the new 2K-Byte Wide Static RAM chips which are **MEM-32K $550**
2716 EPROM compatible. Any 2K byte memory segment can be **MEM-24K $450**
populated with RAM or EPROM (or left empty for use of Address Space **MEM-16K $350**
by another board). Fully expandable to any memory size you will ever **MEM- 8K $250**
need. No special addressing requirements, just solder in extra sockets **MEM- 4K $200**
Extra 2K RAM Memory Chip **$24**
Optional Parallel Printer Port **-P $120**
Optional Calendar/Clock Software available in EPROM) **-T $ 25**
Both options (Disk software mods provided for use of 6522 VIA on printer). **-PT $125**

EXAMPLE USES:

C4P & C8P: Expansion to **4K RAM** of Basic workspace.
Parallel Printer Port — Reserve Serial Port for MODEM
Calendar/Clock Displaying on unused portion of screen.
Space for 5.75K of **Enhanced System Monitor EPROMS.**

All of this on 1 Board, using only one of your precious slots. Software for Enhanced System Monitor capabilities is continuously being developed and improved. As new EPROM Monitors are available, you may upgrade to them for any price differential plus a nominal $10 exchange fee. Another possibility is to fill any portion of the memory with Basic Programs in EPROM for **Power-on Instant Action.** This custom EPROM programming service is available at $25 per 2716 (Includes EPROM). Extra copies at $15 for each EPROM.

C4P-MF & C8P-DF:

Memory expansion to **48K.**
Add 6K **Memory** above BASIC for special software requirements.
Parallel Printer Interface and/or **Displaying Calendar/Clock.**
Add 1.75 K **Enhanced System Monitor ROM.**

C3: Up to **56K of Memory** Expansion — can be addressed for **Multiuser.**
(Optionally, each user can have his own **Dedicated Printer Port).**

C1P,C4P & C8P FLOPPY DISC CONVERSIONS:
Memory/Floppy Board (Includes MI48P1 ROM) ..... **MEM F-16K $450**
C1P-600 Board Adapter & Cable ..... **A600/48 $ 50**
Additional Memory/Printer/Times ..... **(See MEM Board Prices)**
5¼" Drive/Case/Power Supply & Cable to MEMF Board **FD5 $399**

IEE-488 INTERFACES AND SOFTWARE:

The General Purpose Instrumentation Bus (GPIB Controller interface is available for all OSI Computers. Machine code GPIB Drivers are linked to Basic to provide easy control of IEEE-488 instruments which is equal to the best of Hewlett-Packard Controllers and far superior to most others. Basic Commands for Serial Poll, Parallel Poll, IFC Clear, full Local/Remote Control, Respond to SRQ Interrupts, Send Trigger, do Formatted Input/Output, Direct Memory Input/Output and MORE. Interface includes IEEE-488 Ribbon Cable/Connector.

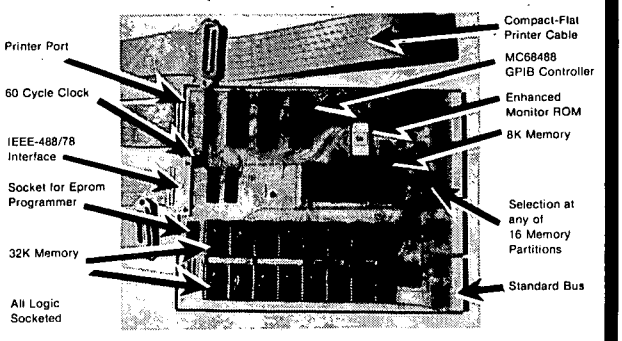
GPIB Controller Interface for C2, C3, C4 and C8 Systems ..... **GPIB 4-488 $395**
GPIB Software for **OS-65D** (Add -8 for 8" or -5 for 5") ..... **GPIB 488-D $ 70**
GPIB Software for **OS-65U** ..... **GPIB 488-U $100**
GPIB Software on two 2716 EPROMS for **ROM** based systems ... **GPIB 488-R $100**
Add Optional **Parallel Printer Interface** to GPIB 4-488 ..... **-P $120**
Add Optional **Calendar/Clock** to GPIB 4-488 ..... **-T $ 25**
Add **2K RAM** to GPIB 4-488 (Specify location, $4000-$BFFF & $D000-$EFFF available) **-M $25**
GPIB Controller for C1P, Includes Software, Clock, All Features of ROMTERMS, & space for 6K EPROM ..... **GPIB 6-488R $395**
Add Optional Parallel Printer Interface to GPIB 6-488R ..... **-P $120**

EPROMS:

C1P ROM with 24/48 Col Display for Series II, Smart Terminal, Line Editing, Corrected Keyboard Screen Clear and More ..... **ROM-TERM II $59.95**
C1P ROM with 24 Col Display. Other ROM-TERM II Features, Disk Boot, and ROM/Disc Basic Interchange ..... **ROM-TERM $59.95**
C4P-MF/C8P-DF Disk warm start, changed IRQ Vector and just flip switch for Serial or Video System with Corrected Keyboard ..... **SYNKEY $39.95**

ENHANCED MONITOR ROMS FOR USE ON GPIB 4-488 & MEM BOARDS:
Expanded Support for **C4P & C8P** Featuring Calendar/Clock, Line Edit, Smart Terminal, Memory Files, Parallel Printer Control, Corrected Keyboard, All Features of ROMTERMS. **Disk Support** with Warm Start and More **MI48P1 $59.95**
Expanded C2 Monitor with Calendar/Clock Software, Hard Disk Boot, Warm Start and Multi-User Control for C2 Systems **MIC2-1 $59.95**

IEEE-488 CONTROLLER INTERFACE



THE GPIB 4-448 INTERFACE BOARD CONVERTS ANY OSI COMPUTER INTO AN IEEE-488 INSTRUMENT BUS CONTROLLER!

BENEFITS — Provides a Sophisticated Instrumentation Controller at very low cost (often saving thousands of Dollars). The combination of IEEE-488 Instrumentation Controller and High Capacity Hard Disk file storage available on OSI Computer systems is available at a fraction of the cost required by the nearest competitor. The IEEE-488 Bus, also known as the GPIB,HP-IB or IEC-625 is the most popular International Standard for connecting instrumentation systems. This 16-line bus is designed to interconnect and control up to 15 instruments at a time. Currently, over 2000 different instruments are available to work on this bus. They include: Plotters, Digitizers, Printers, Graphic Displays, Recorders and a multitude of specialized Test/MeasurementControl Equipment.

EPROM-ABLE — Can be used with a C4-P to create a dedicated IEEE-488 controller.

C2-D MULTIPLE USER SYSTEMS

SAVE — 2 and 3 user Time Sharing Systems are available on the C2-D Winchester Disk Computer at a considerable cost savings from C3 Multiple User Systems. The 3 user C2-D System can be expanded to include a word processing printer, 4 other parallel printers and 3 serial printer interfaces.

COMPATABLE — The special C2-D Multi-User Executive Program is 100% compatable with OS-65U V1.4. The Multi-User Real Time Clock, Memory Partition Control and IRQ Interrupt Management are done on the Micro Interface Memory Board. Thus, the CPU board is not modified and remains in factory condition.

CONVERSIONS — The Up-Grade of your existing C2-D Computer to Multiple User Configuration is also available. Call for details.

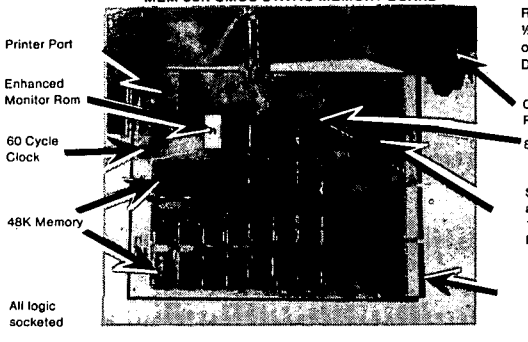
FLOPPY DISK UPGRADES FOR C1P, C4P & C8P

Our Memory/Floppy Board provides easy conversion of 502 and 600 CPU Computers to Floppy Disk Operation. The **MEMF** Board has a floppy disk interface which includes a data separator and the ability to automatically lift the disk drive heads — your floppy disk lifetime will be extended many times. You will retain the cassette interface for your existing software; which can easily be converted to Disk.

This **MEMF-16K** Board is populated with **16K RAM** (50K possible) and has features of the MEM CMOS Static Memory Board with an added floppy interface. The low power memory means extra power supply not required. **ROM Basic** is retained even when Board is populated for 48K Disc Basic. An optional Parallel Printer Port and Real Time Calendar/Clock is on board.

Complete Ready to Run conversion kits with 5¼" or 8" Disk Drives are available.

MEM-56K CMOS STATIC MEMORY BOARD



ULTRA-LOW POWER — By using CMOS Static RAM Memory, the total power consumption is about ½ Amp at 5 Volts when populated for 48K. In fact, most of power is used by the Address Line Buffers and the Data Transceivers.

MULTI-USER — Can be addressed for any of the 16 multi-user memory partitions. The low power and single memory board/partition simplify installation and provide a typical $1400 saving for a 3-user system.

MICRO-INTERFACE
3111 SO. VALLEY VIEW BLVD., SUITE I-101
LAS VEGAS, NEVADA 89102
Telephone: (702) 871-3263

Check with your local Dealer or Order Direct.
Phone orders accepted.
TERMS: Check/Money Order/Master Charge/VISA
Sent POSTPAID ON PREPAID ORDERS.
Foreign Orders: Prepaid only.
Add 5% for handling/shipping.

```
350 702F 205170  ST2    JSR SWITCH     ; 1ST GREATER SO SWITCH
360 7032 A5E3    NEXT   LDA DIR2       ; SET UP NEXT COMPARISON
370 7034 D002           BNE NXT1
380 7036 E6E2           INC DIR+1
390 7038 85E1    NXT1   STA DIR        ; MOVE POINTERS 8 BYTES
400 703A 18             CLC
410 703B 6908           ADC #8
420 703D 85E3           STA DIR2
430 703F B005           BCS UPONE
440 7041 F003           BEQ UPONE
450 7043 4C1770         JMP INIT
460 7046 E6E4    UPONE  INC DIR2+1     ; CHECK FOR DIRECTORY END
470 7048 A5E4           LDA DIR2+1
480 704A C973           CMP #$73
490 704C F017           BEQ EXIT
500 704E 4C1770         JMP INIT
510 7051 A007    SWITCH LDY #7         ; 8 CHARS PER ENTRY
520 7053 E6E0           INC FLAG      ; SHOW THERE'S BEEN A CHANGE
530 7055 B1E1    SW2    LDA (DIR),Y    ; SWITCH THE ENTRIES
540 7057 91E5           STA (TEMP),Y
550 7059 B1E3           LDA (DIR2),Y
560 705B 91E1           STA (DIR),Y
570 705D B1E5           LDA (TEMP),Y
580 705F 91E3           STA (DIR2),Y
590 7061 88             DEY           ; CHAR BY CHAR
600 7062 10F1           BPL SW2       ; MORE TO DO
610 7064 60             RTS           ; SWITCH COMPLETED
620 7065 A5E0    EXIT   LDA FLAG      ; CHECK TO SEE IF CHANGE
630 7067 F003           BEQ OUT       ; NO -- EXIT
640 7069 4C0170         JMP BEGIN     ; YES -- DO IT AGAIN
650 706C 60      OUT    RTS
```

Some Thoughts to Share on 65U

The New One - v1.42

By: Colin Law
P O Box 3819
Auckland
New Zealand

I have spent the past month upgrading from a C3-OEM plus C2-A system to a brand new 230E which includes 65U version 1.42 and a 7 megabyte hard disc. I hope to write further articles on problems encountered (and any solutions I have found) as I become more familiar with the new system.

First, the total installation: 230E with hard disc and one floppy drive, two Hazeltine 'Esprit' terminals, Spinwriter 5530 printer and C-Itoh 1550 dot matrix printer. Operating system is 65U V1.42 with Level 3 time-sharing which results in the terminals being User 0 and User 1. The Spinwriter is device 5, driven through Centronics-type parallel interface and the C-Itoh serial printer is device 8. However, I have built in automatic mapping (on boot-up) of device 8 on User 1 to the device 5 driver. This means that in normal use each user has its own printer available as device 5, this avoids the need for semaphore checking on the printer and until we extend to a third terminal there is no need to use print

LISTING 1

```
1 :REM                <EXAMIN>
2 :REM         A program to search a program
3 :
100 SC$ = CHR$(27)+CHR$(28) : SP$ = CHR$(27)+CHR$(17)
110 FLAG15 : FLAG27
120 PRINTSC$ : PRINT : PRINT "THIS IS < EXAMIN >" : PRINT : PRINT
130 PRINT "You must <LOAD> the program to be  EXAMINed and then"
135 PRINT "OPEN `DATA',`PASS',1 and LIST%1, the program into `DATA'"
140 PRINT : PRINT : PRINT
150 PRINT "LOAD   or   PRINTER  (P5/P8)  or   CONSOLE  "
160 INPUT Q$ : IF Q$ = "L" THEN STOP
165 :
170 PD = 1 : PE$ = "" : PN$ = "" : IF Q$ = "P5" THEN PD = 5
175 IF Q$ = "P8" THEN PD = 8 : PE$ = CHR$(14) : PN$ = CHR$(15)
180 PRINT : PRINT " <EXAMIN> " : PRINT : PRINT : PRINT : X = 1
185 INPUT "SEARCH FOR: "; A$(X)
190 IF A$(X) = "/" OR A$(X) = "" OR A$(X) = " " THEN 200
195 X = X + 1 : GOTO 185
199 :
200 KX = X - 1 : CLOSE : OPEN "DATA","PASS",1
205 PRINT #PD, PE$; ")>> EXAMIN >>> "; PN$;
210 FOR I = 1 TO KX : PRINT #PD, A$(I); "/"; : NEXT
215 PRINT #PD : PRINT #PD : OK$ = "OK" + CHR$(13)
220 FOR I = 1 TO 3 : INPUT%1,A$ : PRINT #PD, A$ : NEXT : PRINT #PD
225 FIND OK$,1 : EOF = INDEX(1)
230 FOR I = 1 TO KX : LN = 0 : LX = 0 : INDEX(1) = 0
235 PRINT "SEARCHING FOR "; A$(I)
240 PRINT #PD : PRINT #PD, ")>>> SEARCH: "; PE$; A$(I); PN$
250 FIND A$(I),1 : IX = INDEX(1) : IF IX > EOF THEN 300
252 INPUT%1,A$ : LL = LEN(A$) : EX = IX + LL
260 INDEX(1) = 0 : IF IX > 80 THEN INDEX(1) = IX - 80
265 SX = IX
270 FIND CHR$(13),1 : TX = INDEX(1)
275 IF TX < EX THEN SX = TX : INDEX(1) = TX + 1 : GOTO 270
280 INDEX(1) = SX
285 INPUT%1,A$ : IF A$ = "" THEN SX = SX + 1 : GOTO 280
290 LN = VAL(A$)
295 IF LN >= LX THEN PRINT #PD,A$ : LX = LN : GOTO 250
300 NEXT I : CLOSE : PRINT #PD : PRINT #PD, "*** END OF EXAMINATION"
320 :
330 END


63999 DEV "E" : SAVE "EXAMIN","PASS"
```

spooling. From User 0 I can print to device 8 (C-Itoh) if necessary, but must first verify that User 1 isn't using its printer. The device 5 semaphore check had to be disabled - with it enabled it assumed that there was only one device 5 and insisted on restricting the printers to print alternately rather than simultaneously !

The manuals supplied are better than previous OSI literature, but it would have been good to see the illustrations instead of 'insert picture here' blanks. Also, it's sad that M/A COM-OSI has inherited the old OSI spelling problems. I realize that in our part of the world we spell some words differently from USA, but these fellows invent their own.... for instance what is 'compatiable' which turns up many times in v1.42 ?

It was first necessary to change several items in our programs to suit the new system. The Hazeltine 'Esprit' terminals are essentially the same as the 1420 I used with the C2 and C3, but with green phosphor and with one or two variations in special codes. Searches included:

- for DEV and DV$: the 230E has device E = hard disc and device A = floppy
- amend a number of WAIT64512,1 references (get a single key input)
- changing several PEEK/POKE routines to new v1.42 flags (allow <,> <:> <CR> as valid INPUT contents)

To search programs for these and other bits and pieces seemed like a long job - it looked as though I would have to print out current versions of every program (over 100) and spend hours marking with red pen. You will guess that I haven't got one of the wonderful 'find-search' type utilities available from some of the PEEK(65) advertisers. Then I decided to let the machine do the work for me and developed (fairly quickly - the need was urgent) my own search system <EXAMIN>. When I get more time (!!) no doubt I'll refine the system, but in the meantime it works quite well . . . .

"EXAMIN"

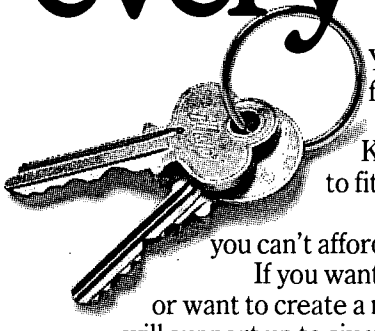
You must have a data file about 30k bytes long - mine is called "DATA", which is easy

```
>>> EXAMIN >>> INDEX(/DATA/A$/FIND/

1 :REM                <EXAMIN>
2 :REM        A program to search a program


)))) SEARCH: INDEX<
230 FOR I = 1 TO KX : LN = 0 : LX = 0 : INDEX(1) = 0
260 INDEX(1) = 0 : IF IX ) 80 THEN INDEX(1) = IX - 80
275 IF TX ( EX THEN SX = TX : INDEX(1) = TX + 1 : GOTO 270
280 INDEX(1) = SX

)))) SEARCH: DATA
135 PRINT "OPEN `DATA',`PASS',1 and LIST%1, the program into `DATA'"
200 KX = X - 1 : CLOSE : OPEN "DATA","PASS",1

)))) SEARCH: A$
185 INPUT "SEARCH FOR: "; A$(X)
190 IF A$(X) = "/" OR A$(X) = "" OR A$(X) = " " THEN 200
210 FOR I = 1 TO KX : PRINT #PD, A$(I); "/"; : NEXT
220 FOR I = 1 TO 3 : INPUT%1,A$ : PRINT #PD, A$ : NEXT : PRINT #PD
235 PRINT "SEARCHING FOR "; A$(I)
240 PRINT #PD : PRINT #PD, "))))  SEARCH: "; PE$; A$(I); PN$
250 FIND A$(I),1 : IX = INDEX(1) : IF IX ) EOF THEN 320
252 INPUT%1,A$ : LL = LEN(A$) : EX = IX + LL
285 INPUT%1,A$ : IF A$ = "" THEN SX = SX + 1 : GOTO 280
290 LN = VAL(A$)
295 IF LN )= LX THEN PRINT #PD,A$ : LX = LN : GOTO 250

)))) SEARCH: FIND
225 FIND OK$,1 : EOF = INDEX(1)
250 FIND A$(I),1 : IX = INDEX(1) : IF IX ) EOF THEN 320
270 FIND CHR$(13),1 : TX = INDEX(1)

*** END OF EXAMINATION
```

LISTING 2 Extracts from TVMENU

```
10 REM This is the main menu <TVMENU>
40 CLOSE:CLEAR
50 FLAG2:FLAG5:FLAG9:FLAG11:FLAG16:FLAG18:FLAG23:FLAG25:FLAG28
70 IFPEEK(18959)<>3THEN63110:REM ENABLE INP$
80 :
100 S$=CHR$(27):SP$=S$+CHR$(17):SU$=S$+CHR$(31):SD$=S$+CHR$(25)
110 SC$=S$+CHR$(28) :PRINTSC$
130 PRINTTAB(15);"Menu follows":PRINT:PRINTTAB(15);"PLEASE WAIT"
135 PRINTSP$;"fi";"*********     Please wait      **********"
140 :
150 LV=PEEK(16317):IFLV=3THENUN=PEEK(55381):GOSUB2000
170 IF UN=1 THEN GOSUB 3000
200 U$="0":IFLV=3THENU$="  User No."+STR$(UN)
210 :
230 A=55919:FORI=3TO5:D$(I)=STR$(PEEK(A+I)+100):NEXT
240 DT$=RIGHT$(D$(3),2)+"."+RIGHT$(D$(4),2)+"."+RIGHT$(D$(5),2)
250 LV=PEEK(16317):BB$="          ":BB$=BB$+BB$+BB$+BB$+BB$
260 PRINTSX$;:FORI=0TO2:PRINTSP$;CHR$(0);CHR$(I);SU$;BB$;BB$:NEXT
270 FH$="**********":FH$=FH$+FH$+FH$+FH$
280 PRINTSP$;CHR$(0);CHR$(3);SD$;LEFT$(FH$+FH$,80);SU$
290 :
300 PRINTSP$;CHR$(3);CHR$(1);SD$;"  TVNZ MENU  "
320 IFLV<>3THEN340
330 PRINTSP$;CHR$(17);CHR$(1);SU$;"  Time Sharing    "
340 PRINTSP$;CHR$(17);CHR$(2);SU$;"  ";U$;
350 PRINTSP$;CHR$(36);CHR$(2);SU$;"LOG:";LG;"%"
360 PRINTSP$;CHR$(36);CHR$(1);SU$;DT$
370 IFLG>80 THEN PRINT" !!";
390 :
400 DIM PG$(60),LN(60),PR(60)
410 READ J:READ A$,PG$(J),LN(J):PR(J)=9:IFJ=60ORA$="XXXX"THEN600
420 IFA$="----"THEN410
430 IFLEFT$(PG$(J),1)="p"THENPG$(J)=MID$(PG$(J),2):PR(J)=0
440 X=INT(J/20)*27:Y=J-(INT(J/20)*20)+4
450 N$=MID$(STR$(J+100),3,2):FF$=SD$:IFJ/10=INT(J/10)ORJ>57THENFF$=SU$
460 PRINTSP$;CHR$(X);CHR$(Y);FF$;[3,"L"] N$;" ";A$;:POKE22,0:GOTO410
470 :
500 T$=LEFT$(" Sorry,"+STR$(PG)+" not ready     ",22)
505 PRINTSP$;CHR$(54);CHR$(1);LEFT$(BB$,22)
510 FORT=1TO6:PRINTSP$;CHR$(54);CHR$(1);
515 FORTI=1TOLEN(T$):PRINTMID$(T$,TI,1);:FORTJ=1TO10:NEXTTJ,TI
520 PRINT" ";:NEXTT
```

New KEYFAMILY™ small business computers for every size business.

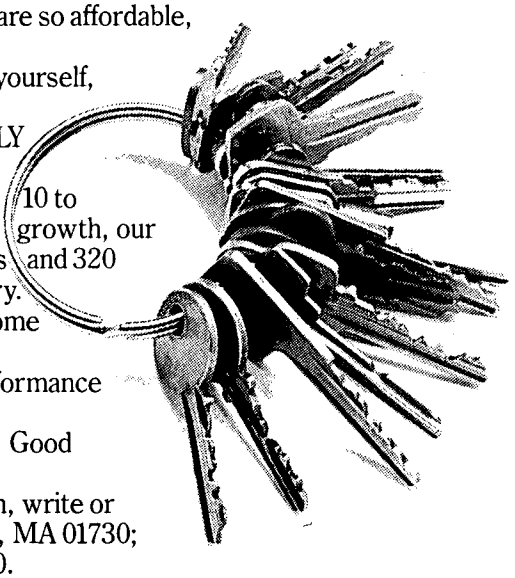


You'll find that a KEYFAMILY computer is the small business system for you, no matter what size your business or your budget.

Whether you are your own boss or the boss of hundreds, the KEYFAMILY line is so complete, you can choose exactly the equipment to fit your needs.

And KEYFAMILY systems are so affordable, you can't afford to be without one.

If you want a single-user system just for yourself, or want to create a multi-user system that will support up to six users, we have the KEYFAMILY system that will fit into your business.



Our multi-user systems allow you to access from 10 to 160 megabytes of hard disk storage. And for even more growth, our network clusters support up to 16 intelligent work stations and 320 megabytes of storage—the largest capacity in the industry.

So, before you commit to any computer system, come in and talk to us. We'll plan a small business system around KEYFAMILY specifically for you. It's the best deal for performance and price you can get anywhere.

That's what we mean when we say, "KEYFAMILY. Good for business."

For the dealer nearest you, or for more information, write or call M/A-COM Office Systems, Inc., 7 Oak Park, Bedford, MA 01730; 1-800-C•A•L•L•O•S•I, in Massachusetts call 617-275-4440.

M/A-COM OSI

KEYFAMILY. Good for business.

"Visit us at Comdex, Booth #1226."

AARDVARK — THE ADVENTURE PLACE

ADVENTURES FOR OSI, TRS-80, TRS-80 COLOR, SINCLAIR, PET, VIC-20

ADVENTURES — Adventures are a unique form of computer game. They let you spend 30 to 70 hours exploring and conquering a world you have never seen before. There is little or no luck in Adventuring. The rewards are for creative thinking, courage, and wise gambling — not fast reflexes.

In Adventuring, the computer speaks and listens to plain English. No prior knowledge of computers, special controls, or games is required so everyone enjoys them—even people who do not like computers.

Except for Quest, itself unique among Adventure games, Adventures are non-graphic. Adventures are more like a novel than a comic book or arcade game. It is like reading a particular exciting book where you are the main character.

All of the Adventures in this ad are in Basic. They are full featured, fully plotted adventures that will take a minimum of thirty hours (in several sittings) to play.

Adventuring requires 16k on Sinclair, TRS-80, and TRS-80 Color. They require 8k on OSI and 13k on VIC-20. Sinclair requires extended BASIC.

TREK ADVENTURE by Bob Retelle — This one takes place aboard a familiar starship and is a must for trekkies. The problem is a familiar one — The ship is in a "decaying orbit" (the Captain never could learn to park!) and the engines are out (You would think that in all those years, they would have learned to build some that didn't die once a week). Your options are to start the engine, save the ship, get off the ship, or die. Good Luck.

Authors note to players — I wrote this one with a concordance in hand. It is very accurate — and a lot of fun. It was nice to wander around the ship instead of watching it on T.V.

CIRCLE WORLD by Bob Anderson — The Alien culture has built a huge world in the shape of a ring circling their sun. They left behind some strange creatures and a lot of advanced technology. Unfortunately, the world is headed for destruction and it is your job to save it before it plunges into the sun!

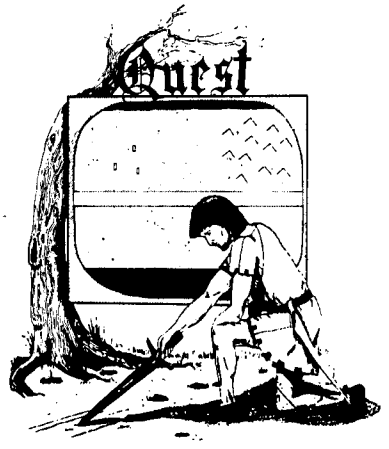
Editors note to players — In keeping with the large scale of Circle World, the author wrote a very large adventure. It has a lot of rooms and a lot of objects in them. It is a very convoluted, very complex adventure. One of our largest. Not available on OSI.

HAUNTED HOUSE by Bob Anderson — This one is for the kids. The house has ghosts, goblins, vampires and treasures — and problems designed for the 8 to 13 year old. This is a real adventure and does require some thinking and problem solving — but only for kids.

Authors note to players—This one was fun to write. The vocabulary and characters were designed for younger players and lots of things happen when they give the computer commands. This one teaches logical thought, mapping skills, and creativity while keeping their interest.

DERELICT by Rodger Olsen and Bob Anderson — For Wealth and Glory, you have to ransack a thousand year old space ship. You'll have to learn to speak their language and operate the machinery they left behind. The hardest problem of all is to live through it.

Authors note to players — This adventure is the new winner in the "Toughest Adventure at Aardvark Sweepstakes". Our most difficult problem in writing the adventure was to keep it logical and realistic. There are no irrational traps and sudden senseless deaths in Derelict. This ship was designed to be perfectly safe for its' builders. It just happens to be deadly to alien invaders like you.



NUCLEAR SUB by Bob Retelle — You start at the bottom of the ocean in a wrecked Nuclear Sub. There is literally no way to go but up. Save the ship, raise her, or get out of her before she blows or start WWIII.

Editors note to players—This was actually plotted by Rodger Olsen, Bob Retelle, and someone you don't know — Three of the nastiest minds in adventure writing. It is devious, wicked, and kills you often. The TRS-80 Color version has nice sound and special effects.

EARTHQUAKE by Bob Anderson and Rodger Olsen — A second kids adventure. You are trapped in a shopping center during an earthquake. There is a way out, but you need help. To save yourself, you have to be a hero and save others first.

Authors note to players — This one feels good. Not only is it designed for the younger set (see note on Haunted House), but it also plays nicely. Instead of killing, you have to save lives to win this one. The player must help others first if he/she is to survive — I like that.

PYRAMID by Rodger Olsen — This is one of our toughest Adventures. Average time through the Pyramid is 50 to 70 hours. The old boys who built this Pyramid did not mean for it to be ransacked by people like you.

Authors note to players — This is a very entertaining and very tough adventure. I left clues everywhere but came up with some ingenous problems. This one has captivated people so much that I get calls daily from as far away as New Zealand and France from bleary eyed people who are stuck in the Pyramid and desperate for more clues.

QUEST by Bob Retelle and Rodger Olsen — THIS IS DIFFERENT FROM ALL THE OTHER GAMES OF ADVENTURE!!!! It is played on a computer generated map of Alesia. You lead a small band of adventurers on a mission to conquer the Citadel of Moorlock. You have to build an army and then arm and feed them by combat, bargaining, exploration of ruins and temples, and outright banditry. The game takes 2 to 5 hours to play and is different each time. The TRS-80 Color version has nice visual effects and sound. Not available on OSI. This is the most popular game we have ever published.

MARS by Rodger Olsen — Your ship crashed on the Red Planet and you have to get home. You will have to explore a Martian city, repair your ship and deal with possibly hostile aliens to get home again.

Authors note to players — This is highly recommended as a first adventure. It is in no way simple—playing time normally runs from 30 to 50 hours — but it is constructed in a more "open" manner to let you try out adventuring and get used to the game before you hit the really tough problems.

ADVENTURE WRITING/DEATHSHIP by Rodger Olsen — This is a data sheet showing how we do it. It is about 14 pages of detailed instructions how to write your own adventures. It contains the entire text of Deathship. Data sheet - $3.95. NOTE: Owners of OSI, TRS-80, TRS-80 Color, and Vic 20 computers can also get Deathship on tape for an additional $5.00.

PRICE AND AVAILABILITY:

All adventures are $14.95 on tape except Earthquake and Haunted House which are $9.95. Disk versions are available on OSI and TRS-80 Color for $2.00 additional.

Please specify system on all orders

ALSO FROM AARDVARK — This is only a partial list of what we carry. We have a lot of other games (particularly for the TRS-80 Color and OSI), business programs, blank tapes and disks and hardware. Send $1.00 for our complete catalog.

AARDVARK - 80
2352 S. Commerce, Walled Lake, MI 48088
(313) 669-3110

Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.

TRS-80 COLOR **SINCLAIR** **OSI** **VIC-20**

to remember. First LOAD your program, then in the immediate mode:

```
FLAG 1 : OPEN "DATA", "PASS",
1 : LIST%1, : PRINT%1,"OK" :
CLOSE
```

This lists the program into DATA in the same format as if it were printed to the screen or printer - i.e. not in tokenized form. Then run EXAMIN and tell it what to search for. Since I run this from User 0 it offers the option of printing to console, printer 5 or printer 8. The first two or three lines of your program are printed, this will usually reveal the program name and other useful reference (that is, if you are in the habit of making your first few lines REMs). EXAMIN will then print all of the program lines containing your selected string. Watch out though - if you're looking for a variable "T" you'll get lines with NEXT, GOTO, PRINT, and many more! Sometimes it is easier in such circumstances to search for combinations such as T=, =T, +T, etc. If the program is too long for your DATA file or if you wish to search only one section it is easy to LIST%1, 1000-9999 or other selected lines. The program EXAMIN is shown as LISTING 1 together with a sample printout from EXAMIN doing some searches on itself.

"START"

Since it is normal every morning to boot-up from hard disc into two-user level 3, I found it necessary to develop an easy way of doing this for operators without program knowledge. The system supplied by M/A-COM OSI requires boot-up to BEXEC*, then selection of multi-user menu, then selection of start-up, etc, etc. As now amended, BEXEC* checks on what level is operational, and if not level 3 then the operator has a clear screen with only "NORMAL START <return>". At this point I can exit with ABORT or STOP to run the BEXEC* menu, but otherwise <return> runs my new program START which is a combination of the relevant sections of MMENU*, LEVEL3, and TSCD07 - the latter being the real works for sorting out level 3 and 7m hard disc. The only question asked of the operator is "RESET TIME/DATE" with the most recently entered time and date displayed. First start up of the day usually involves entering D then the day of the month together with time:

```
525 PRINTSP$;CHR$(54);CHR$(1);LEFT$(BB$,22)
530 QA$="":GOTO600
540 :
600 PRINTSP$;CHR$(54);CHR$(1);SD$;" Enter your choice ";
610 PRINTCHR$(7);:QA$="":FLAG27:INPUT[4,"A"]QA$:FLAG28
615 IFQA$=""THENT$="? ! ? !":GOTO585
620 IFQA$="STOP"THENPRINTSC$:GOTO50160
630 IFQA$="A"THENRUN
640 PG=VAL(QA$):IFASC(QA$))57ORASC(QA$)<48ORPG<0ORPG>59THEN580
650 IFPG$(PG)="TVMENU"ORPG$(PG)=""THEN580
660 PRINTCS$:FLAG23:CLOSE:OPENPG$(PG),"PASS",1:CLOSE:FLAG24
670 IFPR(PG)=9THENGOSUB800
680 IFLN(PG))0THEN700
690 RUN PG$(PG),"PASS"
700 RUN PG$(PG),"PASS",LN(PG)
710 :
800 :              REM PROTECTION WARNING
805 T$="Please ensure that the other user is not using the same files"
810 TT$="PLEASE ensure that the OTHER USER is NOT using the SAME FILES"
815 PRINTSC$:PRINT:PRINT:PRINT:FORI=1TO4:PRINTT$:PRINT
820 PRINTTAB(10);TT$:PRINT:NEXT
830 PRINTSP$;CHR$(10);CHR$(0);"Enter <return> or ABORT   ";SD$;
840 QA$="":FLAG27:INPUT[5,"A"]QA$:FLAG28:PRINTSC$:IFQA$=""THEN870
850 IFQA$="STOP"THENPRINTSC$;"BREAK&NEW":FLAG28:POKE15006,0:FLAG26:NEW
860 IFQA$<>""THENRUN
870 RETURN
1000 DATA00,STAFF FILE EDIT,EDMAFL,0
1001 DATA01,STAFF ANALYSIS,TVRP03,0
1002 DATA02,STAFF REPORT,REPORT,0
1005 :                    REM &etc ..  through to line 1059
1057 DATA57,BACK-UP,TVBACK,0
1058 DATA58,LOG OFF,pLOGLIS,0
1059 DATA59,STANDARD SYSTEM,pBEXEC*,0
1070 :
2000 :                     REM GET FILE
2010 WAIT FOR 99:CLOSE:OPEN"TVLOG0","PASS",1:LF=0
2020 INDEX(1)=9:INPUT%1,EODF:INDEX(1)=20:INPUT%1,BODF
2030 INDEX(1)=31:INPUT%1,RL:INDEX(1)=42:INPUT%1,RN
2040 LG = BODF+((RN-1)*RL) : LG=INT(100*EODF/LG)
2050 LF=99
2060 CLOSE: WAIT CLEAR 99:RETURN
2070 :
3000 :REM                  >>>>> REM RESET C-ITOH PRINTER
3010 PRINT#5, S$;"E";      :REM ELITE PITCH
3015 PRINT#5, CHR$(15);    :REM NON-ELONGATED
3020 PRINT#5, S$;"A";      :REM 6 LINES PER INCH
3030 PRINT#5, S$;CHR$(34); :REM BOLD PRINT OFF
3035 PRINT#5, S$;"Y";      :REM UNDERLINE OFF
3040 PRINT#5, S$;"f";      :REM FORWARD LINE FEED
3050 PRINT#5, S$;"L";"000"; :REM LEFT MARGIN ZERO
3055 PRINT#5, S$;"$";      :REM ASCII CHARACTER SET
3060 PRINT#5, S$;"(";      :REM BIDIRECTIONAL PRINT
3070 POKE15908,60: POKE14457,60: POKE14387,66 :REM RESET PAGING
3080 RETURN
3090 :
50000 REM normal error processing here
63100 :
63110 PRINT:PRINT"Enabling Extended Input Facility - Please Wait":PRINT
63120 RUN"INP$","PASS",63500:REM ENABLE INP$
63130 :
63200 GOTO40  :REM RETURN FROM INP$
63210 :
```

NOTE: The small <p> in front of program names (line 1000-1059) indicates either: protection by way of semaphore checking has been introduced for files OR protection is not necessary. As semaphores are implemented in all files the warning notices will be removed

LISTING 3 <LOGLIS> LOG HANDLER

```
10 REM LOGLIS To report details in TVLOG0
20 CLOSE:CLEAR
30 S$=CHR$(27):SC$=S$+CHR$(28):SP$=S$+CHR$(17)
40 SU$=S$+CHR$(31):SD$=S$+CHR$(25):AS$=" ********** "
100 :
110 PRINTSC$;SU$;:MM$="!!!  IMPORTANT !!!"
120 FORI=1TO11:PRINTSP$;"`";CHR$(I*2);MM$;:NEXT:PRINTSD$
130 PRINTSP$;"yf**   Please ensure that floppy disc drive is empty   **"
140 PRINTSP$;"yk";"Enter <return> to continue ";
150 FLAG27:INPUTQA$:FLAG28
```

8,30,00. The program then continues by running INP$ to bring up extended input and hence the line editor, (later will enable Common Variables - saving variables program to program), on User 1 the program PRTMAP is run to map device 8 through the device 5 driver and both users run TVMENU which is a total system menu with 60 selections, and a log entry is made noting start-up time & date. Thus the operator has entered only <return>, day of month and time to get the system up and running.

TVMENU also shows date, level and user number, plus a % figure showing how much of the log data file has been used (see more about this below). Other items on TVMENU allow for return to BEXEC* (hence system transients and utilities if required) and for review of the log entries.

LOG

This is a data file in which entries are placed by an ever increasing number of programs. Initially it recorded start-up and log-off times, but I have now added log entries for back-up operations and intend to provide for log entries whenever a line 50000 error occurs, i.e. where at present v1.42 programs say "Please log this error" my version will say "This error has been logged". This means that I will be able to review the log at any time and see what has been happening. This will provide a check on what did happen compared with what the operator saw or remembered. There is provision to produce hard copy of the log from time to time and to 'zero' the log data file. When it is time to switch off, the operator selects item 58: LOG-OFF from the menu, program LOGLIS then reminds the operator to check that floppy drive is empty, it then verifies for itself that the drive is empty (by using a reversal of line 50000 error check - it tries to open BEXEC* on DEV"A" and the log-off procedure can continue ONLY WHEN LINE 50000 IS REACHED. If a BEXEC* is found then further warnings are issued.

Listings 2 and 3 show relevant portions of TVMENU and LOGLIS.

BACK-UP

It was essential to prepare some system to back-up the data on the hard disc - and it needed to be done fairly

```
160 IFQA$="STOP"THENFLAG28:POKE15006,0:FLAG26:END:NEW
170 IFQA$="ABORT"THENRUN"TVMENU"
180 IFQA$<>""THENRUN
190 :
200 DEV"A":FLAG23:OPEN"BEXEC*","PASS",1:CLOSE:DEV"E":FLAG24:GOTO50020
210 :
300 FLAG24:DEV"E":PRINTSC$;SP$;"ff";"Enter <return> to log off system"
310 :
330 PRINTSD$;SP$"fs";SD$;"Otherwise enter    ABORT    or    LOG"
340 PRINTSP$;"oo";:FLAG27:INPUTQA$:FLAG28
350 IFQA$="STOP" THEN FLAG28: POKE15006,0: FLAG26: END : NEW
360 IFQA$="ABORT"THENRUN"TVMENU"
370 IFQA$<>""THEN1010
380 :
400 GOSUB500
410 PRINTSC$;SP$;"fi";SPC(9);SP$;"fl";SPC(9):PRINT:PRINT
420 FORI=1TO5:PRINTTAB(60-I*2);"GOODBYE":NEXT
430 GOTO600
440 :
500 A=55919:DT$="":FORI=2TO0STEP-1
510 DT$=DT$+RIGHT$(STR$(100+PEEK(A+I)),2):NEXT
520 FORI=3TO5:DT$=DT$+RIGHT$(STR$(100+PEEK(A+I)),2):NEXT
530 RETURN
550 :
600 GOSUB2000
610 :
700 FORI=1TO5:PRINTTAB(60-I*2);"GOODBYE":NEXT
710 :
800 INDEX(1)=9:PRINT%1,EODF+RL
810 RX=EODF:RG$="LOG OFF":RM$="**"
820 INDEX(1)=RX+FP(1):PRINT%1,[12,"R"]DT$
830 INDEX(1)=RX+FP(2):PRINT%1,[12,"R"]RG$
840 INDEX(1)=RX+FP(3):PRINT%1,[6,"R"]RM$:CLOSE
850 :
900 PRINTSP$;"fi";"Switch off ";SU$;" (1) computer power switch"
910 PRINTSP$"fl";SD$;"and then";SPC(6);SU$;"(2) main wall switch";SD$
920 FORI=1TO5:PRINTTAB(60-I*2);"GOODBYE":NEXT:PRINTSP$;"``";
930 NEW
1000 :
1010 L1$=SU$+"    ENTRY: "+SD$
1020 L2$=SU$+"    REMARKS: "+SD$
1030 L3$=SU$+"  DATE: "+SD$
1040 L4$=SU$+"TIME: "+SD$
1050 :
1060 PRINTSC$: PD=1:PRINT"P5, P8  or Console ";:Q$="C":INPUT[2,"A"]Q$
1070 PRINTSC$: IFQ$="P5"THENPD=5
1080 IFQ$="P8"THENPD=8:FORI=1TO20:PRINT"Check Printer 8 is clear!":NEXT
1090 GOSUB2000:GOTO2200
2000 :                    REM GET FILE
2010 WAIT FOR 99:CLOSE:OPEN"TVLOG0","PASS",1
2020 INDEX(1)=9:INPUT%1,EODF:INDEX(1)=20:INPUT%1,BODF
2030 INDEX(1)=31:INPUT%1,RL:INDEX(1)=42:INPUT%1,RN:INDEX(1)=53
2040 INPUT%1,T$:INPUT%1,T:N=N+1:IFINDEX(1)<BODFTHEN2040
2050 DIM FL(N),FDLB$(N),FP(N+1)
2060 INDEX(1)=53:FORI=1TON:INPUT%1,FDLB$(I):INPUT%1,FL(I):FL(I)=FL(I)-1
2070 NEXT:FORI=1TON:FP(I+1)=FP(I)+FL(I)+1:NEXT:RETURN
2080 :
2200 PRINTSC$ :INDEX(1)=BODF:K=1
2210 INPUT%1,TIME$:INPUT%1,ENTRY$:INPUT%1,RM$:K=K+1
2220 T$=":":PRINT#PD,L4$;:FORI=1TO12STEP2:PRINT#PD,MID$(TIME$,I,2);
2225 IFI<>5ANDI<>11THENPRINT#PD,T$;:GOTO2240
2230 IFI<>11THENT$=".":PRINT#PD,L3$;
2240 NEXT
2500 PRINT#PD,L1$;[12,"R"]ENTRY$;L2$;[6,"R"]RM$
2505 IFINDEX(1)>=EODFTHEN2610
2510 :
2550 IFPD<>1ORK<20THEN2210
2560 FLAG27:INPUT"Enter <CR> to continue ";QA$:FLAG28:K=0
2570 IFQA$="STOP"THEN CLOSE: WAIT CLEAR 99: GOTO2670
2580 IFQA$="ABORT"THENPRINT#PD,"####  ABORT ####   ";:GOTO2630
2590 GOTO2210
2600 :
2610 PRINT#PD:PRINT#PD,"END OF LOG":PRINT#PD:CLOSE:WAIT CLEAR 99
2620 PRINT#PD:PRINT#PD,"Log details up to: ";
2630 GOSUB500:T$=":":FORI=1TO12STEP2:PRINT#PD,MID$(DT$,I,2);
2635 IFI<>5ANDI<>11THENPRINT#PD,T$;:GOTO2645
2640 T$=".":PRINT#PD,"   ";
2645 NEXT:PRINT#PD:PRINT#PD
2650 :
2660 IFPD=5 OR PD=8 THEN PRINT"Z to reset log EODF":PRINT
2665 FLAG27:INPUT"Enter <return> to continue";QA$:FLAG28
2670 IFQA$="STOP"THENFLAG28:POKE15006,00:FLAG26:STOP
```

quickly since a considerable number of reworked programs, together with large data files has been loaded to the hard disc. The floppies we had used on the C2 and C3 were soon out of date. It would be a tedious job copying file by file and in any event some program sizes had been changed. With some 150 files I needed a system which did not require file names, passwords, etc. and v1.42 COPIER was a useful basis to start from. The trouble with COPIER is that it requires you to specify system base address, under level 3 it will copy files only (i.e. from a point 25088 above the base address) and generally requires you to have a calculator handy if you want to back up several megabytes. The back-up system I have developed uses numbered back-up floppies which are identified by a number stored in their DIREC* at index 3000 which is well above the three entries in this floppy DIREC*. The program TVBACK first identifies the floppy in device A, then calculates the system base address relevant to that floppy, then offers options of complete copy of 229376 bytes (68 floppy tracks) from hard disc to floppy, or re-installation from floppy to hard disc, or back-up or re-installation of selected tracks. Yes, 65U doesn't normally operate in 'tracks' but I have used that since floppy track size is the same as sector size on the 7m hard disc and it is a convenient block to work with. The directory program DIR has been modified to show which back-up floppy disc and which tracks contain the back-up of that particular program. (See Listing 4). Since v1.42 files are created only in multiples of 3584 (=floppy track) each program begins and ends on track boundaries. There is provision for 32 back-up floppies, but only 13 in use so far. DIREC* is copied separately to back-up disc 33 and whenever a hard-copy directory is run the DIR program requests disc 33 in slot A and then runs COPYFI to copy the DIREC* from device E to DIRBAC on device A, hence it can be retrieved if a fatal hard disc crash occurs.

Whenever TVBACK is used it makes an entry in the log showing back-up disc number, from & to device, whether copying was completed or aborted and the track/sector numbers copied.

Note: for readers not familiar with CD-7 here are some of the constants: formatted hard disc size 7 311 360, cylinder size 28672, sector size 3584, therefore 8 sectors per cylinder.

```
2680 IFQA$="A"THEN RUN
2685 IFQA$="Z"THEN 3500
2690 :
3000 DEV"E":RUN"TVMENU","PASS"
3010 :
3500 :                    REM ZERO LOG EODF MARKER
3505 RG$="EODF RESET":RM$="******"
3510 GOSUB2000 :INDEX(1)=9 : PRINT%1,BODF+RL :INDEX(1)=BODF
3520 INDEX(1)=RX+FP(1) : PRINT%1,[12,"R"]DT$
3530 INDEX(1)=RX+FP(2) : PRINT%1,[12,"R"]RG$
3540 INDEX(1)=RX+FP(3) : PRINT%1,[6,"R"]RM$: CLOSE : PD=1 :GOTO2660
3550 :
50000 IF PEEK(18176)=23 AND PEEK(18226)=1 THEN 210
50005 :              REM WE WANT ERROR - THAT MEANS NO DISC IN DRIVE
50010 :
50020 PRINTSC$;SU$:FORI=1TO10 :   REM IF DISC IS THERE >> ALARM MESSAGE
50030 PRINTAS$;"PLEASE ensure THAT floppy disc drive IS empty !!!";AS$
50040 PRINTAS$;"Please ENSURE that FLOPPY DISC DRIVE is EMPTY !!!";AS$
50050 FORJ=1TO400:NEXT:PRINTCHR$(7);:FORJ=1TO400:NEXT
50060 NEXT:FORJ=1TO2000:NEXT:PRINTSD$;SC$:GOTO130
50070 :
63999 SAVE"LOGLIS","PASS"
```

LISTING 4 Extracts from DIR

```
915 IF PEEK(RT)=1ANDIT=2THENN$="[----]":TY$="Deleted file":AR$=""
920 IF PEEK(RT)=1 AND IT<>2 GOTO 1000
930 IFDV$="E"THENGOSUB2000          :REM WORK OUT BACKUP TRACK >>>
940 :
950 PRINT #DV, TAB(0);N$; TAB(9);TY$; TAB(16);AR$;
955 PRINT #DV, TAB(22);DA; TAB(33);SZ;
960 SB$="No": IF DA / 3584 = INT (DA / 3584) THEN SB$ = "Y"
965 SL$="No": IF SZ / 3584 = INT (SZ / 3584) THEN SL$ = "Y"
970 PRINT #DV, TAB(42); SB$;"/";SL$;
975 ZC=DA/3584 :PRINT#DV,TAB(49);ZC;TAB(56); :IFDV$<>"E"THEN990
980 IFZC<9THEN PRINT#DV,"BackupDisc 33";:GOTO990
985 PRINT#DV,Z$(1);":";Z$(2);" > ";Z$(3);":";Z$(4);
990 PRINT#DV
995 :
2000 REM SUB TO CALCULATE BACK FLOPPY TRACK
2010 Z(9)=INT(DA/3584) : Z(9)=1+(Z(9)-9)/64 : REM Z(9)=DISC & FRACTION
2020 Z(1)=INT(Z(9)) : Z(2)=9+(Z(9)-Z(1))*64 : REM DISC & TRCK
2030 Z(9)=INT(SZ/3584)+INT(DA/3584) : Z(9)=1+(Z(9)-9)/64 :REM DISC & F
2040 Z(3)=INT(Z(9)) : Z(4)=9-1+(Z(9)-Z(3))*64 : REM DISC & TRACK
2050 FORT=1TO4:Z$(T)=MID$(STR$(Z(T)),2):NEXT
2060 RETURN
```

Yet to come -

Two months before the 230E was delivered I wrote to the local dealer pointing that it appeared that WP-3 wouldn't work in the 65U time-sharing system since WP-3 is based on 65D system. This apparently resulted in telex and telephone messages across the ocean, revealing finally that we would need a different version of WP-3. Now, 3 months later, after using the system for a month we are still waiting to do word processing under the time-sharing system. You would think that 3 months would be sufficient time to get something organized !

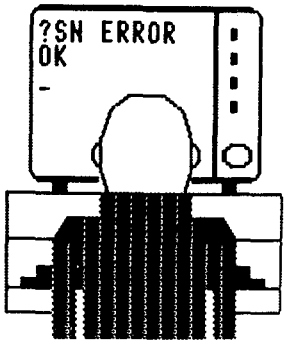
As the system settles down I intend to sort out:

- Files arranged so that back-up disc boundaries coincide with file boundaries, particularly for large data files.
- Files grouped on the hard disc according to `likelihood of change' i.e. some areas of hard disc unlikely to change over several months and don't need regular back-up.
- Extensive use of the log data file to maintain a check on errors and faults.

★

"MIDNIGHT HACKER"

Modified & Submitted by:
Earl Morris
3200 Washington
Midland, MI 48640



To The Midnight Hackers

IT'S TERRIFIC!
(A Happy User)

By: Al Peabody

My friend Dick McGuire calls being in the computer business "Dealing with the dissatisfied." And with good reason. If an installation is smooth, everything works fine, you the dealer or programmer never hear from the customer again. But if something goes wrong, if the software is not well suited to the particular way the customer does HIS payroll, or if a memory chip fails at a young age, you will hear fom him every single day!

```
10 REM MIDNIGHT HACKER
20 REM
30 REM by S.A. Smith 1981
40 REM
50 REM OSI/UK USER GROUP
60 REM Photo And Modifications By
70 REM Earl Morris
72 REM
75 REM Chuck Stanford
78 REM was here too!
80 VI=53458:LL=64
90 IFPEEK(57088)>128THENVI=53446:LL=32
100 CU=VI+LL*4+3:T$(1)="SN ERROR"
110 T$(2)="?OM ERROR":T$(3)="NO CHANCE"
120 FORS=1TO32:PRINT:NEXT
130 PRINT"To The Midnight Hackers:"
140 FOR L=1 TO 14
150 FOR C=1 TO 18
160 READN:IF N=0 THEN N=32
170 IF N=1 THEN  N=187
180 POKEVI+L*LL+C,N
190 NEXT:NEXT
200 POKECU,95:T$(4)="?TM ERROR"
210 FOR TE=1 TO 4
220 T$="RUN":GOSUB610:FOR DE=1TO2000:NEXT
230 T$=T$(TE):GOSUB 280:FOR DE=1 TO 1000:NEXT
240 T$="OK":GOSUB 280:FOR DE=1 TO 3000:NEXT
250 GOSUB 370:FOR DE=1 TO 1200:NEXT:GOSUB 400
260 NEXT TE
270 GOTO 210
280 FOR  CH=1 TO LEN(T$)
290 POKE CU+CH-1,ASC(MID$(T$,CH,1))
300 NEXT CH
310 FORDE=0 TO 500:NEXT
320 FOR C=0 TO 9:W=CU+C
330 A=PEEK(W-LL):POKEW-2*LL,A
340 A=PEEK(W):POKEW-LL,A
350 POKE W,32:NEXT
360 POKECU,95:RETURN
370 B=VI+4*LL+9:POKEB-LL,161:POKEB-LL+1,161
380 POKEB,177:POKEB+1,175:POKEB-LL-1,177:POKE
    B-LL+2,175
390 RETURN
400 B=VI+4*LL+9:POKEB-LL,32:POKEB-LL+1,32
410 POKEB,32:POKEB+1,32:POKEB-LL-1,32:POKEB-L
    L+2,32
420 RETURN
430 DATA,221,148,148,148,148,148,148,148
440 DATA148,148,148,148,217,148,148,222,,
450 DATA140,,,,,,,,,,,,149,173,,139,,,140,,
460 DATA,,,,,,,,149,173,,139,,,140,,,,,,,,
470 DATA,,,149,173,,139,,,140,,,,,221,195
480 DATA135,135,197,222,,149,173,,139,,,140
490 DATA,,,,202,,,,,199,,149,,,139,,,140,,,
500 DATA224,130,,,,,143,225,149,227,22,139
510 DATA,,220,148,158,148,148,200,,,,,201
520 DATA148,215,158,148,223,,210,135,135
530 DATA135,135,135,1,1,1,1,1,1,135,135,135
540 DATA135,135,207,209,128,128,176,161,1,1
550 DATA1,1,1,1,1,1,161,178,128,128,208,136
560 DATA,,1,1,1,1,1,1,1,1,1,1,1,,,143,209
570 DATA154,161,1,1,1,1,1,1,1,1,1,1,1,1,161
580 DATA154,208,209,128,128,1,185,1,1,1,1,1
590 DATA1,1,1,186,1,128,128,208,,,,1,185,1
600 DATA1,1,1,1,1,1,1,186,1,,,
610 POKEVI+14*LL+14,32:POKEVI+14*LL+15,32
620 FOR DE=1 TO 100:NEXT
630 POKEVI+13*LL+14,128:POKEVI+13*LL+15,128
650 FOR  CH=1 TO LEN(T$)
660 FOR DE=1 TO 800:NEXT
670 POKE CU+CH-1,ASC(MID$(T$,CH,1))
680 NEXT CH
690 POKEVI+13*LL+14,186:POKEVI+13*LL+15,187
700 FOR DE=1 TO 100:NEXT
710 POKEVI+14*LL+14,186:POKEVI+14*LL+15,187
720 GOTO 320
    OK
```

★ ★ ★

This note is intended to tell the other side of the story; a computer installation which works fine. A customer who is satisfied. It is a real life story, about real people and a real machine, a C3-OEM.

The Location

The business involved is Free State Press, a quick print shop in Annapolis, Maryland. Annapolis is a small city, but it is the state capital of Maryland, so the need for printing services is perhaps greater than in other towns of the same size. Free State Press is a progressive print shop with high speed duplicating and copying equipment, as well as complete photocomposition and offset printing facilities.

The Problem

The owner/operator of the shop, Jim Martin, knew for months he needed a computer to help him with billing and job scheduling. "I went to the altar a half dozen times but never said 'I do,' until my monthly billing job became just impossible," Martin reports.

At this point, Martin gave in and bought a computer, with great fear and trembling. On the day the machine was installed, he was almost as nervous as an expectant father. Would it really work? Would it be able to send out accurate statements to his hundreds of customers? Would it be easier to enter transactions, charges and payments, into the computer than to write them onto ledger sheets? And would the computer take care of his growing word processing and custom letter-writing business as well? And how about job scheduling? And cash flow projection?

The Solution

That's a lot of questions. Fortunately, after 3 months, some of the answers are in, and those answers are "yes!" Let's examine some of the processes involved in making this a successful installation.

First of all, Jim Martin is a realist. He did not expect his computer to do everything in his office immediately. He was able to sit down with a consultant and decide which job was the most important to him. They decided that if the computer could take over the billing process immediately, it would be worth the expense. The other functions planned for later start-up would be extra benefits, but not crucial to the initial success of the installation.

This is a key factor. If a computer can't justify itself with one single application, think very carefully before buying. If it can, even though you are sure you will use it in many ways as time goes by, you can be sure you will be happy with your new electronic business partner. In the case of Free State Press, the Accounts Receivable/Billing cycle was the key.

The Solution to this particular problem came in the form of a C3-OEM, a dual-floppy, 48K system capable of running both M/A-COM OSI's powerful OS-65U and Digital Research's popular CP/M operating systems. This allowed Martin and his consultant to select from a broad range of software to do the jobs needed immediately and anticipated for the future.

The Accounts/Receivable problem was handled by a package called "Charge," provided by DBMS. Charge uses OS-DMS file structures and utilities to maintain a file of customers

THE DENVER BOARD
(You've heard so much about)
WHAT IS IT ? ? ? ?

- **60k of 2 MEG. MEMORY**
  use programs already written in OS65-U, or write new programs using the extra memory space.
- **A 6502 MICROPROCESSOR**
  the fastest general purpose eight bit processor on the market.
- **AN AUXILIARY 48 PIN BUS**
  allows connecting special purpose cards to the DENVER BOARD without interfering with normal bus operation.
- **A BUS ARBITRATOR**
  allows up to 16 simultaneous users, running different jobs.
- **INDEPENDENT RESET**
  use the break key on the STANDARD terminal to reset one user without interfering with other users.
- **LITTLE OR NO SPEED DEGRADATION WHEN EXTRA USERS ARE ADDED.**
  mini speed with a micro.
- **AUTO-BOOT**
  switch selectable auto-boot-floppy or hard disk.
- **A SIMPLE UPGRADE**
  multiple DENVER BOARDS run standard OS65-U level 3 programs.

EXAMPLE #1

timeshare separate disk areas 74 meg. = 14 users (5 meg. each) auto-boot switch on, keep utilities on floppy for programing uses.

EXAMPLE #2

dedicated printer, use aux. bus for device #5 card and print all output from this user, other users will be unaffected by long printer outputs.

ORDER NOW

LIST: $1295.00
Introductory Price ONLY 995.00

TERMS:
Send 10% deposit, BAL. C.O.D., or send full amount and save C.O.D. and shipping charges.

OR SEE A PARTICIPATING DEALER

NAME: _____

ADDRESS: _____

CITY: _____ **STATE:** _____ **ZIP:** _____

PHONE: ( ) _____

☐ **Please send** _____ **boards, enclosed is $** _____

☐ **I do not wish to order now, but send more information. I may be interested in** _____ **boards in the future.**

DEALER INQUIRIES INVITED

DBI, inc.
P.O. Box 7276
Denver, CO 80207
Phone:
(303) 364-6987

who have been granted credit, showing their current, 30, 60, 90 and over 90 day balances and other information. The system also maintains a file of transactions for the month, showing customer number, date, item (such as invoice number, or check number if it is a payment).

It is into this transaction file that information is entered as sales are made and payments received. Of course, all of this is transparent to the user. He or she merely selects "enter sales" or "enter payments" from the main "menu" of choices, then answers the questions which appear on the screen.

By the end of the month, the transaction file has grown to reflect all of the credit activity of the business for the month. At this point, just before monthly statements are printed out, the menu selection "monthly A/R report" is made, and the computer produces a very important report. It starts with the prior balances listed in each customer's file, then applies all the transactions for that customer in the transaction file for the month, to produce the new account status for the customer. Each customer is given one line on the report, a "snapshot" of where his account stands.

Martin takes this report and looks it over for mistakes. He does not expect the computer to make a mistake--these machines add and subtract quite well! -- but it is certainly possible that sometime during the month an employee has made a mistake, entering the wrong amount or the wrong account number; and now is the time to catch it.

If there are mistakes, it is a simple matter, using M/A-COM OSI's file editor program, to correct the error in the transaction file.

Then, when the month's transactions are correct, the program is run which prints out monthly statements for all customers. This program also automatically applies payments to the oldest balances in each account, and ages the balances one more month. It also applies a service charge, and prints out any one of several messages on the bottom of each statement, depending on such things as the age and amount of the oldest balance, whether a payment has been received in the last month, etc. An advertising message may also be printed on all statements if desired.

Jim Martin reports that the use of the computer and "Charge" have greatly speeded up the process of putting out monthly statements, decreased the rate of errors in monthly statements, allowed payments received just before statements are mailed to be reflected in the statements, and made his statements friendlier. He also likes the total control he has when statements go out and what they look like.

The Future

One major reason for selecting the C3-OEM computer was its ability to run both OS-65U and CP/M. Martin has purchased two major CP/M products, T/Maker II and WordStar with MailMerge. He uses T/Maker II to generate tables in which each row represents a job in process, each column a date. It is easy to schedule the many steps through which a printing job must go, and to adjust the schedule as jobs proceed and problems develop.

WordStar and MailMerge are used to write custom letters, with each letter addressed to a different person, and variable imformation "dropped" in place in the letter where it is needed. This process is a direct replacement for a job done earlier on a Magnetic Card typewriter, but with WordStar's full-screen editing and simple variable insertion procedure, it is much faster and easier.

...And Even More Jobs

Jim Martin is hardly through thinking up things to do with his computer. His payroll is still done manually. His accountant still does his books, using information which Martin must supply, though now some of this information is generated by "Charge."

The large inventory of many different types of paper, inks and other supplies is manually maintained.

But first of all, Martin would like to "turn around" his "Charge" program, to do the same thing for A/P that it now does for A/R. "Perhaps we should call it 'retreat,'" he says in jest.

"Actually, I've tried another Accounts Payable program, and it just wasn't as easy to use,

and didn't produce as useful, timely information as "Charge," so why not just turn it around, revise it to write checks instead of statements, and even give me a "cash needed projection" instead of aged receivables report? My consultant-programmer is working on it right now, and I expect to have it up an running within a month or so."

And after that? "Oh, there's plenty more this machine can do for me. Why, sometimes we actually turn it off for a half hour at a time!"

CEGMON:

A Note from the Originators

It seems that some users are a little confused about certain aspects of CEGMON, the monitor EPROM for OSI video systems. As the originators of the system (our surnames, Chkiantz, Elen & Graves, are the CEG in CEGMON) we would like to take this opportunity to clear things up.

CEGMON is today the "de facto" standard monitor in UK video systems. We also believe it to be the best. Since its release in September 1980, only one bug has been located, which was rapidly corrected. We, therefore, believe it to be bug-free, and, so it seems, do users. Regrettably, CEGMON has been somewhat misrepresented by the US agents, Aardvark. We have asked them to correct some of the mistakes but have not succeeded to date. As Roger Olsen is a very nice person, we're sure that it's merely an oversight. We also wonder a little about royalty payments.

First, the Aardvark ads called it something strange --C1U was it?-- rather than CEGMON, although the instructions call it CEGMON correctly. It is NOT the same as the Dutch monitor, which we think they called the C1S. This may have confused some people.

Second, the ads totally erroneously claimed that CEGMON solved the Garbage Collector problem. Of course, this is a function of BASIC 3, and would not be affected, even in a much more clever monitor than ours. You have to modify the BASIC 3 code to do the job properly. Indeed, we wonder why you people still get so upset about Garbage Collectors. The OSI/UK User Group published the correct (later improved) Garbage Collector modification in 1979! Almost every UK user has a faultless GC today because they have bought new ROMs which contain our code. All these weird (and often wrong) fix programs are merely playing with the problem and NOT solving it. Microsoft got their math wrong, and that's what you correct. We told Aardvark many times about this misleading statement but they never changed it!

Third, for some reason Aardvark copied our manual into their word processor and photocopied the output. As a result you get a rather fading matrix-printed manual with all the page numbers referring to the "original UK manual" pages. Thus, when it tells you a certain example is on page n, you won't find it there, because THEIR page numbers are different from ours! We suggested that they supplied our manuals which would have been very cheap for us to print for them: they never took us up on it. Our manual is a proper 20-page printed and typeset manual with programming examples, etc. There is also a handy reference card with useful commands and locations, missing from the US version.

We apologize to US users who might have been misled by these statements and put off by the documentation. For reference, this is basically

what CEGMON does:

* A screen editor for use with BASIC or Assembler programs; uses second cursor to copy any line on screen into the 'new line' with keyboard entry at any time.

* A revised keyboard routine giving typewriter-like response, true ASCII key values (except delete = $5F rather than $7F for OSI compatibility) and access to most graphics, plus 'single-key' BASIC commands.

* A completely new screen handler which offers program or keyboard access to clear-screen, home cursor, multiple scrolling, protected screen areas, cursor control and separate 'text window' with its own clear command, plus true delete and printing from top of current 'window'.

* A full machine code monitor with many of the functions of the Extended Monitor with the exception of a disassembler.

* Disk bootstrap --unlike some other UK 'new monitors' we kept this, so you can still boot a disk. CEGMON is designed for ROM BASIC systems but can be linked into 65D with no hassles, giving even a 65D 3.2 system more facilities than 65D 3.3.

* I/O vectored through RAM for easy linking to your own I/O routines.

* Compatible design -- gives the maximum practical compatiblity with SYNMON as the major routines are in the same place. Most existing software which runs under SYNMON will run under CEGMON with little or no change, and if changes are needed they are simple and obvious.

Several versions of CEGMON are available for different display and machine types. All the Superboard versions are easily fitted: the C4 version requires an extra decoding chip (full instructions are given). A new version is available for the new Superboard. If users upgrade their displays, the existing CEGMON can be used by POKEing the screen window RAM locations; or the chip can be exchanged.

I hope this clears up any misconceptions users or potential users may have.

Richard Elen
UK User Group
London, England

LETTERS

ED:

When the New Brunswick Telephone Company (NBTel) decided to offer a time share computing service, a host of methods for implementing such a service were considered. Following an extensive evaluation of the systems available and their individual benefits, the decision to purchase a local area network (LAN) using UCSD Pascal from Becterm Inc. [418/837-5894] was made.

Our basic networking system as supplied by BECTERM was one C3-B (single user) with RAM at D000 to EFFF, BECTERM's UCSD V2.1 Network Software, two BC-INET support system (powers up to 8 BC-INETS), fourteen BCINET CPU's with 63K of RAM (printer option available), one 9 track tape drive and one 8" floppy disk drive.

The expansions which are presently available from BECTERM, include OS 65U V1.30 to run currently with U.C.S.D. Pascal, up to 31 BC-INETS per node, up to 2 nodes per Network and a GT option (3.2 MHz).

The basic network system supports UCSD PASCAL and UCSD FORTRAN 77. When expanded, it will also support OSI's disk basic (OS-65U V1.3).

BECTERM provided us with the necessary system utilities with its Network system.

1. Account maintenance source software which allows the system manager to create new accounts and maintain existing accounts.

2. Log on source software which acts as the gateway to the Network along with system mail.

3. Conversion software to convert OSI's floppy disk format to BECTERM's disk format.

4. A Network time and date spooler.

5. A Network printer spooler which will allow up to four printers to be located on a front end computer (any one of the BC-INET computers).

BECTERM is presently considering expansions and updates to the Network. At the present time there are plans to upgrade the network to UCSD PASCAL Ver. IV, include a nine track mag. tape spooler and upgrade OS 65U from V1.30 to V1.45.

We at NBTel have had this Network system operational for approximately seven months and have uncovered a few minor flaws with the system. The ones we encountered are:

1. W.P. 3.3 software requires changes in the operating system (we have decided to use a UCSD word processor).

2. OS 65U disk access is slow, but comparable (in speed), to floppy disk access.

3. No UCSD Pascal Ver 4.0 available.

4. Disk definition is cumbersome.

The problems associated with the network have not been insurmountable and we feel that the reliability that we have received has outweighed the problems encountered.

H.B. Evans
St. John, New Brunswick

MR. Evans:

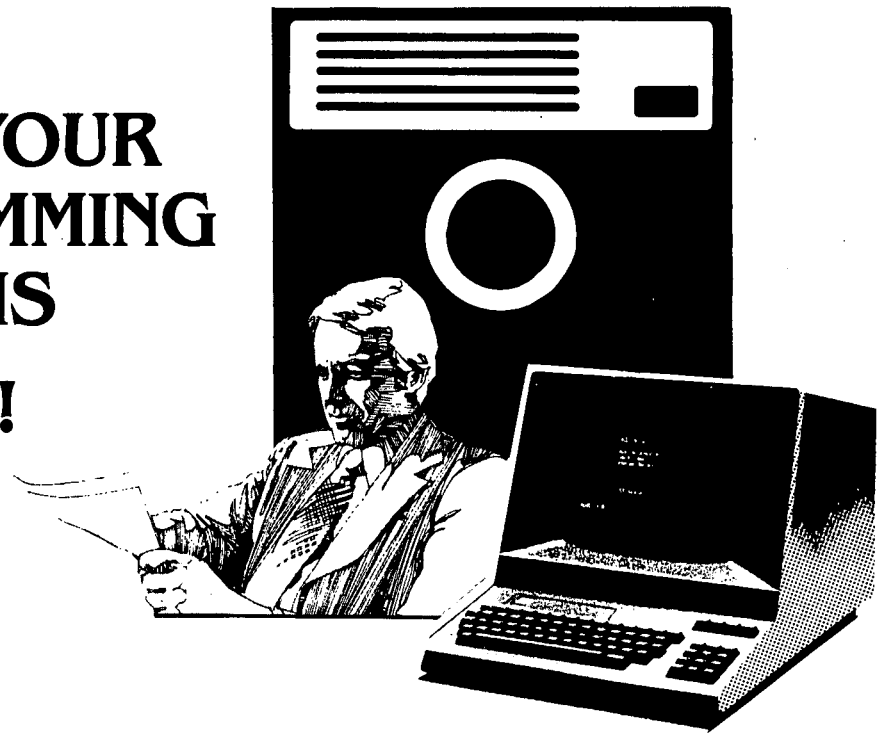
We have found 65U disk access on the 74MB disk far and away the fastest in the industry. I wonder why it is so slow in your system....any ideas?

We would love to hear more details on the problems you encountered and how you solved them.

Al.

* * * * *

TACKLE YOUR PROGRAMMING PROBLEMS HEAD ON!



If operating bugs and "unsolvable" coding problems have got you dodging a big project, here's help . . . Quality software products that'll let you tackle those problems and move on to something easier.

At Software Consultants, we know the kinds of problems you often face with OSI gear, so we designed our software accordingly. Most of our products are modifications and/or extensions of OSI packages, but some are complete replacements. All are easily extensible for your own custom applications. We feel they're the finest OSI software products around – and the most reasonably priced.

PRODUCT ROUNDUP

OS-65D V3.2 DISASSEMBLY MANUAL

A super-complete manual that has it all. 50 pages of disassembly listings, complete and clear comments on all lines, 10 pages 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.

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

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 the printer and temporarily storing it on hard disk. Written in super fast machine language. Interfaces with serial and parallel printers. Only $69.95.

FIG FORTH UNDER OS-65U

The only other language available for your 65U system. You get lots of extras, like terminal oriented editor, printer and terminal tools, and much more. Under multi-user , runs BASIC simultaneously too. Unbelievable at $89.95.

VIDEO ROUTINE

This convenient extension for OS65D V1.2 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.

GRAPHICS RESOLUTION BOOSTER

An ingenious piece of hardware that increases 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 C4/8P 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 postage & handling, we'll send you the source code you choose.

Foreign orders please add $10 postage fee for air mail. Dealer inquiries welcome. Tennessee residents add 6% sales tax. COD orders will incur a $1.50 handling charge. We ship via UPS, so please include street address with your order.

Write or call us today with your order, or ask for our free product catalog and get all the details.

SOFTWARE CONSULTANTS
HARDWORKING *software*

6435 Summer Avenue
Memphis,TN 38134
901/377-3503

ED:

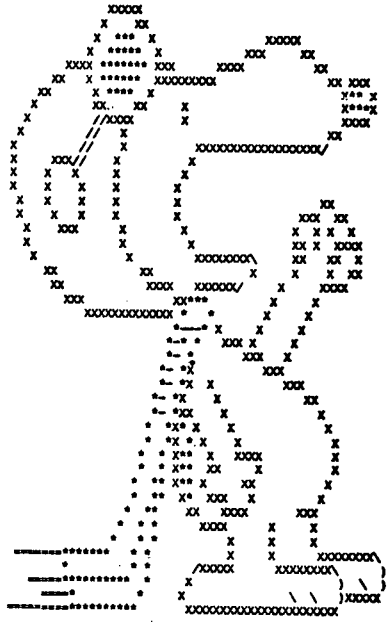
When PEEK (65) first came out, you had many articles and correspondence on OSI C3s. I appreciated that because there are many magazines on the market that cover OSI's small systems but virtually none that I know of that cover the OSI C3s. It appears as though PEEK (65) is beginning to follow that trend also. Must be that there aren't that many C3 systems out there.

Possibly I'd be wasting everyone's time by mentioning that there is an outfit called HEURISTIQUE that is making a collection of the CP/M public domain software on OSI C3 8" format. Their collection now ranges from volumes 1 to about 50 with a few copies through the 60s and 70s. The best part, as a sideline they are selling a copy service such as Lifelines is doing but for the OSI formats. The price is the same as that from Lifelines which is currently at $8.00/ Volume. There is one thing to watch out for however. Some volumes are "squeezed" and/or "archived" and since it is only a copy service there is no comment of that on the volumes. Also, unsqueezing them and unarchiving them becomes your responsibility. HEURISTIQUE has accumulated the squeeze/unsqueeze - archive/ unarchive programs all on a separate volume. The address is: HEURISTIQUE; P.O. Box 30386; Portland, OR 97230.

The big thing in public domain software in this neighborhood seems to be pictures. One of the biggest, which I don't have and probably won't get, is a commercial airline flying west over the Golden Gate Bridge. I saw the picture and it is about an eight foot square picture and I was told it took about four ribbons to print it. Personally, I find the utilities software much more interesting and very useful.

I've generated some software for myself to transfer packets of data between OS-65D files and CP/M. In doing this I've discovered a few OSI "gotchas" that I suppose I should have known about but forgot or just didn't know about. For those who wish to know, read on. OS-65D version 3.0 loads memory from a diskette file starting at 317E-H. CP/M looks for data at 0100-H. When CP/M is booted up; it boots in the same location as OS-65D or at 2200-H. If data is moved down to 0100-H before CP/M is booted, you may find strange data at 2200-H on. Also, data will be altered if the prom monitor is used to analyze data that is is resident in memory at 012D-H. A "gotcha" that both CP/Ms DDT and OS-65Ds EXTENDED MONITOR has, that may not be readily apparent, is that when data is moved from one location to another, it could conceivably eat itself up if the start of the new location is in the middle of the old location. These are a few of the problems that gave me continual headaches and maybe will avoid some for you. Maybe someone smarter than myself or at least someone with more patience than I have, will devise a method of copying OS-65D files directly from the diskette to CP/M files on to diskette and visa/versa. It would be nice to exchange OSI BASIC with other BASICs but I guess that other than CP/M users, there is not too much excitement to do such a thing.

Arthur Gores
Portland, OR 97220



CURSE YOU, MERGENTHALER!

Arthur:

We have more C1-C2 articles only because more of them are submitted. If C3 users will submit articles, we will print them!

With two computers, you can transfer ASCII basic programs between 65D and CP/M, but due to different disk access techniques they must be extensively modified to run if disk files are used.

Thanks for the Snoopy graphic!

Al.

* * * * *

ED:

This letter is prompted by Wm.K. Groover's article on rounding numbers in the September '82 issue of PEEK (65).

When I first got my C1, I wrote a checkbook recon program. I then discovered how awkward BASIC is when it comes to printing dollars and cents. To my surprise, I found when a number is converted to a string by the STR$ operation, the result includes a space being added at the left end of the string. Thus, STR$(A) where A-99 yields '_99', where _ is a space. And, LEN (STR$(A))= (number of digits in A)+1.

Thanks in part, to a fellow C1 owner, I now have a fairly straightforward subroutine for converting an integer to a dollar-and-cents figure for printing:

```
10 REM  A=INTEGER FORM OF
   AMOUNT
20 C$=RIGHT$(STR$(A),2):
   D=INT(A/100)
30 D$=RIGHT$(STR$(D),1)
40 PRINTD$;".";C$
```

Suppose A=789. STR$(A) would equal'_789'. LEN(STR$(A)) would equal 4. In line 20, C$ equals the two right-most characters of STR$(A), or '89'. D equals the integer value of A/100, or 7. Line 30 produces a D$ of '7'. Now we get '7.89' when we print in line 40.

We can print a column of figures whose decimal points all line up by padding with spaces. The easiest way to do this, is to anticipate the greatest length D$ will ever be. Then, rewrite line 30 to handle it, such as:

```
30 D$=RIGHT$(STR$(D),3)
```

In this instance, we can print a D$ from one to three digits long. The previous example of D=7 would produce a D$ equal to '_7'. D=67 would print as '_67', and D=567 would print as '567'. D>999 would only print the three right-most digits.

If A is ever <10, it must be stored with a leading zero so

Business System Users

Get Mainframe Software features with
"Computer Business Software"
"CBS"

the **INSTALLABLE** Accounting Package
for Floppy Disk and Hard Disk Systems -
Single or Multi User

Featuring: Accounts Receivable with Sales Analysis
Inventory with Product Analysis
Order Entry with Invoicing
Accounts Payable with Checkwriting
General Ledger with Financial Statements
Payroll with Labor Distribution

- Thoroughly Documented
- Already Installed
- Fully Supported
- Ready Now

Call **1-800-843-9838** for your free documentation kit, price schedule and sales brochure.

MICROSOFTWARE INTERNATIONAL

3300 South Madelyn, Sioux Falls, South Dakota 57106
"Mainframe Software for your Micro"

that there will be no space after the decimal. Example: if A=5, then RIGHT$(STR$(A),2) would equal '_5'. The printout would be'0._5'. So, we would have to do the following:

```
25 IFA<10THENC$="0"+RIGHT$
   (STR$(A),1)
```

Bruce Showalter
Abilene, TX 79601

Bruce:

Would this work?

```
5   SP$= "                    "
10  REM A= INTEGER AMOUNT IN
    PENNIES
20  A$=MID5(STR$(A),2):REM
    STRIP LEADING SPACE
30  A$=LEFT$(A$,(LEN(A$)-2))+
    "."+RIGHT$(A$,2):REM
    INSERT DECIMAL
40  A$=LEFT$(SP$,(10-LEN(AS)))
    +A$:REM PAD TO 10 CHAR
    WITH BLANKS
```

Of course, the blank is the sign digit, and will be a "-" for a negative number - perhaps we should make line 20

```
20  IF LEFT$(A$,1)=" " THEN A$
    = MID$(A$,2):GOTO 20
```

To strip any leading blanks but leave " " alone. Or how could we put parentheses around negative numbers?

Al.

* * * * *

ED:

I have a C4P with dual 5 inch disk drives, and an MX 80 printer. For software I have OS-65D V3.2, OS-65D V3.3, WP6502 V1.2, DQ-Justify, DQ-Secretary and WP6502 V1.3. I also have several excellent utility programs from AARDVARK, including the machine Code EDITOR.

I have given the said versions V3.3 and V1.3 respectively each a fair trial with less than satisfactory results. My problem is, that although I am not a skilled touch typist, I nevertheless have certain fingering habits that I carry over from my acquaintance with the typewriter, and I still have occasion to use the typewriter when filling out printed forms or questionnaires. Consequently, I am still seeking the best and highest degree of compatibility between my fingers, the hardware and the software.

I purchased the V3.3 DOS because of its normalized keyboard feature, but I am disappointed in it because, in its EDIT mode, I do not like the unconventional mid-line behavior of the <RUB-OUT> key. I am also disappointed because it will not allow me to type the @ character, nor does it have the handy automatic repeat function. Finally, its COPIER is now a separate 2-track program, and I do not regard the extra features of V3.3 to be worth the extra space on my disks.

In lieu of V3.3 I have installed SYNKEY, an eprom from MICRO-INTERFACE, that gives me the normalized keyboard that I want, (although I still can't type an @ character in the direct mode).

With regard to the WP6502 V1.3, I am disappointed with that because it forces me to use the <ESC> key in lieu of <RETURN>. This conflicts with my fingering habits and confuses me even when using other software. Instead, I am using WP6502 V1.2 with DQ-Secretary and DQ-Justify enhancements. With this word processing combination, I have been able to change the embedded command codes, and I can type @ and #, and I use the <ESC> as my embedded command marker.

So, for all future revisions of disk operating systems, word processors and EDITOR programs, here is my WISH LIST:

1. Let <RUB-OUT> always be exclusively a destructive back-space.

2. Make all characters repeat automatically, when any key is held down.

3. Let <SHIFT><0> = @, and not line abort.

4. In the word processor, leave <ESC> available for use as the embedded command marker.

5. Make # available from the keyboard so that it need not remain the embedded command marker.

6. In the EDIT mode, both in BASIC and in the word processor, move the cursor non-destructively using the <CTRL> key in combination with the following: > (to right), < (to left), U (up) and D (down). (The last two should be effective only in BASIC when the cursor is at the left margin.)

7. In the direct mode use <ESC> for entering the EDIT mode.

8. In general, pattern the EDIT mode after the AARDVARK machine code EDITOR.

9. Insofar as possible, coordinate the disk operating system, the BASIC language, the word processor and the BASIC EDITOR so that each key consistently performs the same function.

So far, I have made some progress toward these goals. I am continuing to use OS-65D V3.2 and WP6502 V1.2, and in addition to the SYNKEY eprom, I have cut my right <SHIFT> key switch loose and wired it in parallel with the left <SHIFT> key contacts. This is something OSI should have done for us long ago, but I consider myself fortunate that I have been able to do it now.

With my present modifications and enhancements, I have partially succeeded in attaining some of the goals which I have listed, but the coordination is far from perfect. I pray that OSI and DQFLS will heed my pleas and move in a direction to gratify all my wishes.

Carl M. King
Sarasota, FL 33579

* * * * *

ED:

I've finally figured out how to implement that AUDIT TRAIL feature that's hidden away in EDMAFL (Edit Master File Utility/DMS). I wrote a note to you, Al, addressed to your office, asking for help (from the masses) to figure out how to avoid the disc error I kept getting (when toggling on the Audit trail feature in line 126 (F6=K2 instead of K1); since I am the impatient sort, and since I can really use this feature for the EBBA mailing list, I set upon the task to figure out why the program bombed out... and found the answer (and could have hit myself in the head for not noticing it sooner). In line 5530 and 5540, there are TAB's set for 90 columns; it bombed out because I was using an 80 column printer! The enclosed changes fix this problem in use with the Centronics 737 printer (which is the printer I use because it's lots faster than the ITOH for this sort of stuff).

There was, however, another error in line 5520, which

originally read:

```
5520 PRINT#AD,"REC #: ";FC$
     (K1);TAB(53);"FIELD:
     ";FDLB$(FPTR);
```

where FC$(K1) is the old field contents, and we don't want that where the record # should be, so I changed that variable to ;RPTR; -- which now correctly identifies the physical record number (and since the EBBA Mailing list 'Membership Numbers' are the same as the physical record numbers, that kills two birds with one stone (pardon the pun)). However, nothing is ever perfect, as line 5500 needs to have something done to it too (I don't off-hand know what, though - any suggestions, Al?) it now reads 5500 IFF6=K1 OR S$="/" OR S$=FC$(FPTR) GOTO 5550 (or bypass the audit trail)... problem is that if I hit a carriage return (in order to leave the field contents as it is), it ALSO prints the trail, which it isn't supposed to do. I think perhaps this might be changed to... OR S$="" GOTO etc...? [Yes. - Al]

Each time that I use this editor program, in order to change an entry in EBBA's mailing list data file, when the audit trail is switched on (by answering a prompt in the beginning of the program), the change is reflected on hard copy. The printed information will read:

```
DMS EDITOR    (DATE)    REC#:
(NBR)    FIELD:  (FIELD NAME)
OLD (Contents)
NEW (Contents)
```

As you realized from my sometimes aggravating correspondence, I've had constant problems with the second disk driver (DEV B), and though it suddenly started to operate again two weeks ago, it also conked out again a few days ago. Frustrating beyond belief, since the DMS software and EBBA data file is very difficult to use with a single drive.

This morning, I decided time had come to do a major maintenance job on the drives (remember, these are old GSI's), since anything I did (in my opinion) couldn't mess things up worse anyway. So I undid the drives from the frame, pulled them out, cleaned them thoroughly (cripes those contacts were BLACK!), found NUMEROUS loosened screws, VOLUMINOUS amounts of filth (no wonder it didn't work- all that mess probably blew around in there from the draft created by the driver motors!), but although I'm not sure what the culprit really was (probably a combination of things), I do suspect the catch on the door (it locked too tightly), so I adjusted the whole thing so it visually appeared like the working unit (DEV A), and lo and behold- now it works fine! (Wonder seriously for how long!) I'm still planning to get new Shugart double density/double sided drives anyway, since I need the storage space. In any case, it's a much cheaper solution than buying a new system or a more expensive OSI.

I have a very special question for you... I want to do something else to EDMAFL, but lacking the programming experience, I don't want to louse it up.

What I need is some way to load a field with the same constant in a new file. For example, let's suppose that I've created a brand new DMS file and I want to load into that file from an existing file, all the records, where let's say, the contents of one field is "6"... Since there currently is no way to prepare the new file, i.e. to preload that contents of '6', I want to program something that does it for me. I realize that even what I've tried to program is very limited, i.e. I still will have to run through the records (amend) but at least I don't have to do a lot of typing.

Question is, could this work, and do you have a better idea for improvements... (surely you must!)?

EDMAFL

NEW LINES:

```
128 PRINT:INPUT"WANT TO LOAD A
    FIELD (Y/N)";WL$
129 IF WL$<>"N" THEN GOSUB
    7000

7000 REM
7010 PRINT:INPUT"NAME OF
     FIELD";LF$
7020 PRINT:INPUT"NEW
     CONTENTS";CF$
7030 PRINT"NEW CONTENTS OF";
     LF$;"WILL BE";CF$
7040 PRINT:INPUT"Yes or
     No";QF$
7050 IF QF$<>"Y" THEN GOTO
     7000
7060 IF QF$="Y" THEN RETURN

1121 IF WL$="Y" AND IF FDLB$
     (FPTR)=LF$ THEN S$=CF$:
     GOTO 1142
```

I got the idea from the DT$ business... question is, will it work? Wouldn't it be nice if you could devise a way to by-pass the EDMAFL altogether and create another editor whose sole function is to input this type of data (I tried it with SEEDLD but that program is so screwed up it's unbelievable... it should be noted that when I got this DMS originally, it was a dealer's copy of an OSI copy (without serial number I might add, I've even written OSI about that at the time but in THOSE days they didn't answer correspondence... and it was full of bugs; I paid the then list price for it... now it works in a fashion, i.e. all except the key file create but I don't need that anyway- I use KYUTIL instead, which is a super program!)

Enclosed is an amendment to EDMAFL (OSDMS 9/79) which can serve as a guideline for many other projects. I think the "Further Explanation" under the code is self-explanatory, but if not, please let me know. Basically, what happens, is that IF field 2 (AOU#) holds one of the numbers on the look-up listing, then the corresponding content (for that AOU number) will automatically enter field 3 (Common Name). If the inputted AOU#, however, does NOT match any of the numbers in the table, then field 3 will wait for an input- that's because the last entry (not shown) after the table is:

```
1899 GOTO 1130
```

Preliminary & incomplete AOU lookup table from 1500-1899 in 'EDMAFL'.

```
1500 REM SUBR FOR AOU LOOKUP
     TABLE
1510 REM
1520 IF FDLB$(FPTR)="COMMON
     NAME" THEN GOTO 1530
1530 AOU=VAL(FC$(2))
1540 IF AOU=3220 THEN S$=
     "SHARP-SHIND HAWK":
     GOTO 1142
1550 IF AOU=3870 THEN S$=
     "YEL.BILL CUCKOO":
     GOTO 1142
1560 IF AOU=3880 THEN S$=
     "BLK.BILL CUCKOO":
     GOTO 1142
```

Code that gets you to this section of the program:

```
1122 REM
1123 IF FDLB$(FPTR)="COMMON
     NAME" THEN GOTO 1510
1130 INPUT S$:GOSUB5500
```

Further explanation: Field 2 is 'AOU#'/field 3 is 'Common

Name'. The contents of field 2 determines the contents of field 3.

Each species of birds is assigned a species number by the American Ornithologists Union (AOU), thus if field 2 (AOU)=3220, then this amendment to the program automatically enters the name 'Sharp-Shind Hawk' into S$ for field 3. Note that the government forms for which this lookup table is designed limits the 'Common Name' to 16 characters. The full name of this species is Sharp-Shinned Hawk.

Frederick S. Schaeffer
Jamaica, NY 11435

Fred:

Thanks for an interesting letter. Please realize that though your double-sided Shugarts will double your disk capacity by allowing you to specify the "back" sides of the disks as drives "C" & "D", you will NOT be able to use the double density feature, since 65U does not contain double-density (more bytes per track) software. Your total capacity for files will be 1,003,520 bytes.

Your program "looks" like it would work, but would require you to "edit" every record in the file to do it! However, the ONLY way to be sure a program will work is to test it, thoroughly, on dummy data.

If you want to do the job automatically, have a look at a program which uses DMS master files (such as EDMAFL) and see how they open up the files and get the BODF (beginning of data), EODF (end of data), NR (number of records), and RL (record length), set up arrays of field labels (FD$(X)), field contents (FC$(X)) and field offsets (FP(X)). Then if you want to enter, say "PAID" in field 6 of each record, do something like:

```
FOR CT =1= TO NR
INDEX<1> = BODF + (CT-1) * RL
+ FP(6)
PRINT%1,"PAID"
NEXT X
```

And of course, TEST IT on a dummy file (do it, then try to edit the file, print it in a report, etc.).

Al

* * * * *

ED:

After enthusiastically typing Jack Watts' annual calendar (PEEK (65), August, 1982) into my souped-up C1-P, I came to the sad realization that the program had a severe bug that resulted in printing a calendar in which the first day of each month was indeed the 1st! I went over the program carefully and made sure that I had not mistyped anything. NOPE! No errors on my part. I decided not to worry, because surely (like other magazines) you would correct any errors in the September issue; so when the September issue arrived, I searched in vain for a correction to the annual calendar program.

Perhaps my problem has something to do with the compatability of the program to the C1. For example, I cannot use variables like those in lines 410 (viz. LD%) and 3310 (viz. WK%). I should be able to substitute regular variables here, shouldn't I? Anyway, somewhere there is an error.

Thank you for the continued support for OSI products. Your newsletter is especially helpful to me, since I am a language arts teacher at a high school and rely almost soley on OSI computer products for work in my classroom.

Samuel W. Shive, Jr.
Jacksonville, FL 32244

* * * * *

ED:

Mr. Shive has found the problem, I believe. Evidently OSI 65U 8K BASIC-IN-ROM which is what I used to write the program, treats WK% as WK=INT (etc.), that is, it amputates the decimal regardless of the value. My new system is Microsoft Basic Version 5 which rounds to the nearest integer - UP OR DOWN - depending on the value of the decimal.

Here is a listing of the changes that should solve the problem in how Basic handles %. Incidentally, the changes also include a modification by Mr. Willis Cook, that enables right hand justification for the numerals which greatly improves the presentation.

Basically, the changes before line 3510 concern the % problem and afterwards are right hand justification.

```
410 Y=YR-1700:LD=INT((Y+100)
    /400):LD=INT((LD+(Y/4))-
    INT(Y/100))
3310 WK=INT(DA/7):B=DA-(WK*7):
     IF M=2 THEN E=E+ED
3720 IF M=1 OR M=4 OR M=7 OR
     M=10 THEN A$(L)=RIGHT$
     (STR$(A),2)
4020 IF A>9 GOTO 4110
4030 IFM=20RM=50RM=80RM=11
     THENPRINTTAB(W*3+22)B$(L)
     ;:L8=A:GOTO4410
4040 IFM=30RM=60RM=90RM=12
     THENPRINTTAB(W*3+46)C$(L)
     ;:L9=A:GOTO4410
4050 GOTO 4310
4310 IF A<10 THEN PRINT TAB
     (W*3-2)A$(L);: L7=A: GOTO
     4410
4320 PRINT TAB(W*3-2)A$(L);
     : L7=A
6710 TB=2
11010 TB=2
```

Jack K. Watts
Honolulu, HI 96815

* * * * KYUTIL NOTICE * * * *

Several people have written to ask about compatability of KYUTIL with OS 65U ver >=1.3; here are the changes:

1. Change the first line of both "MOVE" and "BACK" to read:

```
1 FLAG 15
```

2. Make the following changes to "MCCRE" some disks may not have this program and in that case the changes should be made to "MACCRE":

```
1230 DA=FA: GOSUB62000: GOSUB
     1680: REM BRK DN DA OF
     FILE
1250 DA=LN: GOSUB62000: GOSUB
     1680: REM BRK DN LN OF
     FILE
62000 REM SET LENGTH AND
      ADDRESS ON SECTOR
      BOUNDARIES
62010 X=INT(DA/3584): IF DA-
      X*3584<>0 THEN
      DA=(X+1)*3584
62020 RETURN
```

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:

PEEK (65)
P.O. BOX 347
OWINGS MILLS, MD. 21117

SUBSCRIPTION RATES FOR 12 ISSUES (ONE YEAR), EFFECTIVE WITH THE JULY, 1981 ISSUE. ALL RATES QUOTED IN U.S. DOLLARS.

PLEASE FILL OUT AND RETURN WITH CHECK OR MONEY ORDER.

( ) $15.00 ENCLOSED. U.S. (MARYLAND RESIDENTS ADD 5% SALES TAX.)
( ) $23.00 ENCLOSED. CANADA AND MEXICO, 1ST CLASS, SURFACE.
( ) $35.00 ENCLOSED. SOUTH AND CENTRAL AMERICA. AIR MAIL.
( ) $35.00 ENCLOSED. EUROPE. AIR MAIL.
( ) $40.00 ENCLOSED. ALL OTHER. AIR MAIL.

NAME _____ STREET _____

CITY _____ STATE _____

ZIP CODE _____ COUNTRY _____

PLEASE SEND THE FOLLOWING BACK ISSUES. I ENCLOSE:

( ) $2.00 EA. U.S. SURFACE. (MARYLAND RESIDENTS ADD 5% SALES TAX.)
( ) $2.50 EA. CANADA AND MEXICO. SURFACE.
( ) $3.00 EA. SOUTH AND CENTRAL AMERICA. SURFACE.
( ) $3.00 EA. EUROPE. SURFACE.
( ) $3.50 EA. ALL OTHER. SURFACE.

Vol 1. 1980

( ) JAN #1	( ) FEB #2	( ) MAR #3	( ) APR #4
( ) MAY #5	( ) JUN #6	( ) JUL #7	( ) AUG #8
( ) SEP #9	( ) OCT #10	( ) NOV #11	( ) DEC #12

Vol 2. 1981

( ) JAN #1	( ) FEB #2	( ) MAR #3	( ) APR #4
( ) MAY #5	( ) JUN #6	( ) JUL #7	( ) AUG #8
( ) SEP #9	( ) OCT #10	( ) NOV #11	( ) DEC #12

Vol 3. 1982

( ) JAN #1	( ) FEB #2	( ) MAR #3	( ) APR #4
( ) MAY #5	( ) JUN #6	( ) JUL #7	( ) AUG #8
( ) SEP #9	( ) OCT #10		

INDEXES ARE INCLUDED IN THE JANUARY & DECEMBER 1981 ISSUES.