

MUG Newsletter # 17 - December 1981

\*\*\*\*\*

A SHORT COURSE IN COMPUTER NUMBERSby Burks A. Smith of DATASMITH  
Box 8036, Shawnee Mission KS 66208

Those of us who are familiar with the internal workings of computers and computer languages that are not as "high level" as BASIC use some number systems that are unfamiliar to most people. In fact, we get so comfortable with these numbers that we tend to assume that everyone who has a computer knows what we are talking about. Of course, most people have no idea what a hexadecimal number is and many MUG members may not be so sure either. They are probably too embarrassed to ask. Hopefully, this little article will shed some light on the subject.

Most people today use the "decimal" number system, a method of writing numbers that is based on ten. This system is quite logical when you realize that Humans have ten fingers, but surprisingly has only been in use in Europe since about 1300, when it replaced Roman Numerals. The decimal system is a positional system, with the rightmost position representing ones, the next position tens, etc. These positions represent powers of ten and a digit in a position indicates how many of the position's value are in the number. Thus 256 is two hundreds, five tens, and six ones. There are ten symbols for numbers in the decimal system: 0-9. Ten has no symbol of its own, so it is written using two symbols. So far this is pretty simple stuff, right?

Now there is nothing special about a number system built on ten. It isn't any "better" than other number systems, and if we had twelve fingers we would probably be using a number system based on twelve. A computer has no fingers at all. In fact, a modern digital computer really only recognizes two states, one state is when current is flowing or a voltage is present, and the other is when it is not. Since we are limited to two logical states, the "natural" number system for digital computers is one that is based on two. This "binary" number system follows all the same rules as the "decimal" number system. The only difference is that there are only two symbols: 0 and 1. It is a positional system just like decimal, except the positions are powers of two, not ten. Thus, the rightmost digit is still ones, but the next digit is twos, the next fours, the next eights, etc. The value of any position is two times the one on its right. Thus, in binary, the number 101 is one four, zero twos, and one one. This adds up to five in decimal. This is such a simple system even a computer can do it. There are no "addition tables" to memorize like humans do. There are only three possible addition problems:

0+0=0  
0+1=1  
1+1=10

Note that since there is no symbol for two in the binary system, the rule is "bring down the zero and carry the one" when you add 1+1, just like when you add 5+5 in decimal.

Inside your computer's memory, all data is represented in some sort of binary form. In computer talk a "bit" is one binary digit that can take on a value of either 0 or 1. In most micros, memory is arranged so that the smallest unit that can be accessed in a single operation is eight of these bits, and this unit of eight bits is called a "byte." "Bytes" in memory are accessed according to their "address", which is represented inside the processor as two bytes, or a sixteen bit binary number. While binary numbers are very convenient for a stupid computer that doesn't know any more

than the difference between zero and one, they are not so convenient for people. When you write them down they take up a lot of paper, and it is hard to read all those zeros and ones. For this reason, you don't see binary numbers very often unless you have a computer with a front panel that displays binary numbers in the form of lights.

Because binary numbers are so hard for us humans to deal with, most operating systems and monitors use the "hexadecimal" number system to communicate binary information. Hexadecimal numbers are not any more "natural" for a computer than they are for people, so the translation between binary and hexadecimal is done within the program. As you may already know, hexadecimal numbers use a base of sixteen. This means there are sixteen different symbols for digits instead of the ten for decimal or the two for binary. The first ten symbols are the digits 0-9 and, for convenience, the first six letters of the alphabet (A-F) are used for the rest. The number sixteen is represented as 10 (pronounced one-zero) following the rules of all of the other positional number systems. Therefore, from right to left, the values of the positions are the powers of 16, or (in decimal) 1, 16, 256, 4096, etc. The hexadecimal number 3B means three sixteens and eleven (B) ones, making a decimal equivalent of  $3*16+11$  or 59.

Since sixteen is a power of two, hexadecimal numbers can represent binary numbers more easily than the decimal system can. Each hexadecimal digit is the equivalent of four binary "bits", since 1111 Binary equals F Hexadecimal equals 15 Decimal. In order to represent an 8-bit "byte" only two hexadecimal digits are required, and each digit can be rather easily translated mentally (with some practice) to four binary bits. This is really the only reason hexadecimal numbers are used. It is simply for the convenience of people.

In actual practice, it is seldom necessary to convert between number systems when using an operating system such as MDOS or a monitor such as the one provided by Vector Graphic. The computer speaks to you in hexadecimal and expects you to reply in hexadecimal. Most of the time you are dealing with memory addresses and single bytes, so the decimal equivalent is irrelevant because it doesn't tell you anything anyway. Under MDOS version 4, the user area begins at address 2B00 Hex. This is a nice "round" hexadecimal number, so knowing that this is 11008 in decimal and 0010110100000000 in binary isn't much help.

If you want to do anything more than program in BASIC, you will need to become familiar with binary and hexadecimal number systems. It really isn't hard to do. All it takes is some practice and fiddling with the computer. Incidentally, if you want to learn more about programming your computer on its own level, I recommend the book: 8080 ASSEMBLY LANGUAGE PROGRAMMING by Lance Leventhal. Get this book even if you have a Z-80 processor, because both the Micropolis ASSM assembler and the Vector Graphic ZSM assembler use the Intel 8080 conventions for mnemonics.

.....

MICROPOLIS BASED SYSTEMS (PART 2)

by Buzz Rudow

LATAH

The LATAH computer system is a microcomputer based data processing system which includes CPU, memory, floppy disk drives, optional rigid disk drives, video display, I/O interfaces, optional desks and printers, and a variety of system and application software.

The CPU is an 8085 operating at 3 megahertz. It runs on the S-100 bus, has a 3K-byte PROM and up to 56K of RAM. Seven S-100 slots are available for expansion.

Disk drives, both floppy and rigid, are Micropolis. Floppy options run from a single MOD-1 to four MOD-IIs. The rigid disk is available in 9, 27, and 45 megabyte versions. A 50 megabyte digital tape backup is also an option.

The Video Display is an I/O device with 25 lines x 80 characters. It supports the full ASCII set (upper and lower case plus special symbols) on a high resolution 12" CRT. The keyboard contains an alphanumeric typing array similar to that found on a standard office typewriter. It has an eighteen key numeric keypad with standard 10-key adding machine layout. The entire system may be locked through the keyboard.

The LATAH has two RS-232 serial ports which supports baud rates of 110 to 9600. Three parallel ports, with handshaking, are also included.

For hardcopy, printer selections include a 110 cps model from Epson, a 150 cps model from Texas Instruments, and one from Florida Data which runs at 900 cps.

System software included in the base price of the floppy system consists of a resident monitor, Micropolis PDS (MDOS, LINEEDIT, DEBUG, ASSM, and the other utilities) Micropolis Basic, and five application programs (discussed below). The rigid disk system includes OSM, a multi-terminal, multi-tasking operating system, which provides high-speed, keyed access to large files.

The application programs, tailored for the LATAH, include the following:

INTERRELATED GENERAL BUSINESS PACKAGE  
 General Ledger, Payroll, A/P, A/R, & Inv.  
 CUSTOMIZED MUSIC STORE  
 CUSTOMIZED FURNITURE STORE  
 CUSTOMIZED WHOLESALE DISTRIBUTOR  
 CUSTOMIZED LAND MOBILE RADIO DEALER  
 CUSTOMIZED COOP  
 CUSTOMIZED FEED ANALYSIS & LEAST COST  
 CUSTOMIZED MANAGEMENT JOB COSTING & PAYROLL  
 CUSTOMIZED REAL ESTATE  
 WORD PROCESSING & TEXT EDITOR  
 CUSTOMIZED ACCOUNTING  
 CUSTOMIZED PROCESSING & MANUFACTURING  
 CUSTOMIZED OIL & GASOLINE JOBBERS  
 CUSTOMIZED POTATO PROCESSING  
 CUSTOMIZED PHARMACY  
 CUSTOMIZED COMMODITIES SETTLEMENT  
 DATA BASE MANAGEMENT  
 BASIC EXPANSION MODULE

For further information, contact LATAH Business Computers, 5114 Gage, Boise, Idaho 83706, 208/375-4305.

.....

#### SYSTEM MEMORY

by Buzz Rudow

If you don't like to use the MEMEND statements, you can use your video display memory and system RAM for assembly language routines. Of course, not all members have this type of memory available. You also have to be careful that it isn't used for anything else at the time.

It's kind of amusing to execute Systemation's SORT/A in my SOL's video display memory when some unsuspecting sole is in the room. Talk about Star Wars presentations! The screen goes wild. I believe people think the computer is about to explode. I just do a CLEAR-SCREEN when I'm through and go on my way. The program gets written over when you do any other output to the screen. That means you need to reload SORT/A each time you need it, but the savings in contiguous memory space is worth it.

Both SOL and VG also have a 1K system RAM. Both systems use the upper portion of this RAM during Micropolis execution. Therefore, you can't load a

full 1K program like SORT/A. Something smaller is OK, and it will stay there and not get written over.  
 .....

#### DEBUG & TRACK DENSITIES

by Bart Bailey, Box 704  
 Mildura 3500, Victoria, Australia

#### DEBUG GENERATION PROGRAM VS. 4.0

It was found that debug could not be generated at page address F0, although the handbook allows that address. Investigation with DEBUG found that a compare instruction was given as F0H which excludes this address from being generated. The problem can be solved by loading DEBUG-GEN into memory and changing the byte listed below using the ENTR command.

Address  
 2B6BH FE F0 ; CPI F0H  
 2B6BH FE F1 ; CPI F1H new byte

Now goto the debug generation program using the APP command. When the sign on message appears carry on as per handbook section 4-92.

#### DISK TRACK CONFIGURATION

To modify "RES" to obtain different track densities ie., 35, 40 and 77 tracks, change the bytes listed below using ENTR command.

Address 0884H XXH Disk Drive 0  
 0885H XXH " " 1  
 0886H XXH " " 2  
 0887H XXH Disk Drive 3  
 XXH is the number of tracks in hex.  
 Note...35 decimal = 23 hex  
 40 " = 28 "  
 77 " = 4D "  
 80 decimal = 50 hex

After changing the track byte resave RES per handbook section 2-33, steps 5 to 8.  
 .....

#### COMMERCIAL SOFTWARE

##### PROFESSIONAL TAX RETURN PREPARATION

Taxpro is a set of programs for the accountant, CPA, or lawyer who produces many tax returns for his clients. The system has been designed to prompt the user, ask for input and to operate on the results accurately and in accordance with the current tax law. Every effort has been made to insure ease of operator input and to minimize the necessity for side-calculations.

Any or all of the following Federal forms are produced by TaxPro: Form 1040, pages 1 and 2; Schedules A, B, C, D, E, ES, F, G, R, RP, SE, TC and U. Forms 1116, 1116 Schedule A, 1310, 2106, 2119, 2210, 2210F, 2440, 2441, 3468, 3903, 4136, 4137, 4255, 4625, 4726, 4797, 4874, 4952, 4972, 5329, 5695, 5884 and 6251 are produced.

Where both Taxpayer and Spouse have sole proprietorships, employee moving expense or employee business expenses, separate schedules are produced when appropriate. Up to 10 salary items, 30 interest receipts, 30 dividends and 20 estates, partnerships or Sub-S corporations are accepted. Any of these may have associated capital gain or loss, investment credit, etc. Up to 3 Schedule C's and 3 rental properties are accommodated. An optional overlay program will allow processing of about 125 dividend, interest, salary, stock sales or other items.

State returns will be available December 15th for California, New York and New Jersey for both residents and non-residents. Resident returns are

available for Florida (intangibles tax), Pennsylvania, Maryland, Virginia, North Carolina, Indiana, Connecticut and Ohio. Other state returns will be made available where there is interest.

A diagnostic report is provided. This features useful information and makes sophisticated tax planning more feasible. Data produced by the system can be stored for generating taxpayer questionnaires with comparative data for the following year or for batch processing. A collating and billing guide is also produced.

An inexpensive run-time package, available from Systemation, inc., is required for operation.

Minimum hardware requirements are 56KB computer with Microplis DOS. CP/M and IBM personal computer versions will soon be available. Double density disks are suggested. Two disk drives are standard but the system can be configured for single drive systems where the considerable sacrifice of processing efficiency is acceptable.

Straight-line and declining balance (125,150, DDB) are computed with operator override allowed for other methods. Conventional half-year depreciation is allowed. Gains and losses under Secs. 1231, 1245, 1250-5 are computed. Personal portions of real estate tax and mortgage interest on rental properties are carried forward to Schedule A where necessary. All exceptions for Form 2210 are accommodated. Returns can be prepared on government forms, commercial substitutes or on plain paper using acetate overlays. The latter method is highly recommended. Where plain paper processing is preferred, the user has the option of continuous fan-fold forms or single sheet feeding.

The software is licensed for a single computer and to a single end-user. The cost of the license is \$2000. Programs for state returns are \$300 each, other than New York and California, which are \$500. A tutorial session is available in my office for \$300 for the day. The optional overlay and subsequent year proforma program will each be \$200. Annual updates will be available for 80% of the then-current new user price.

For further information, contact Martin C. Rothstein, CPA, 21 E. 40th ST, NY NY 10016, 212/683-5310.

#### TEXTWRITER

If you want to write a letter or other text type document, what do you use on the Micropolis system? You can use LINEEDIT or EDIT/S to construct the document, and then the PRINTP command to print it. This isn't bad for short documents. Long or fancy documents require something a bit more powerful, though.

I've been using TEXTWRITER. While it isn't as flexible as some of the \$300-\$500 word processors, it gets a lot of work done, and some of the things it does, the "big boys" can't do. This newsletter is produced with it, for example. It does work in conjunction with an editor such as LINEEDIT or EDIT/S. TEXTWRITER is available from Organic Software, 1492 Windsor Way, Livermore CA 94550 (415) 455-4034, for about \$125. Remember to specify the Micropolis MDOS version. Otherwise you'll get CP/M on Micropolis.

What follows is an excerpt from the 100+ page operations manual. In subsequent months I'll go into detail on the functions and commands of TEXTWRITER.

#### A LITTLE BACKGROUND

Word processing consists of three basic functions. First, there is text editing - the process of entering or modifying text stored by the computer. Second, there is text formatting - the process of arranging text on a page with proper margins and

justification. Third, there is the actual printing of the document.

There are two different types of computer programs that perform word processing; they are either "screen processors" or "text formatters". A screen processor performs the formatting function of word processing simultaneous with the editing function, so that the results of any editing changes can be seen directly on the terminal. A text formatter, however, relies on a separate editing program to make text changes, and formats the text as it is printed on the printer.

A text formatting program, such as TEXTWRITER, is superior in many ways to a screen processor. The major shortcoming of a screen processor is that the text is formatted while it is entered by the operator. The printed output is limited to what can be displayed on the screen because the margins are limited to the width of the terminal display. This means that when printing a form letter, the name and address of the addressee must be manually entered into the proper place in the original since the length of the name may vary. This also means that to use one common paragraph in several documents, that paragraph must be manually copied into each different document.

A second major shortcoming of a screen processor is that a special terminal is required. Such a terminal must have cursor addressing and other special functions. A text formatter, on the other hand, works with any "dumb" terminal.

All of the above disadvantages are overcome by TEXTWRITER. When printing form letters, it allows names and addresses to be inserted from a separate file or from the keyboard - as the file is printed. It allows standard paragraphs or sections to be inserted as the document is printed. When printing reports, manuals, or other documents, it automatically produces a table of contents, alphabetized index, and also places footnotes. Furthermore, it works with any computer terminal and printer.

When combined with a powerful editing program, TEXTWRITER'S performance is superior to any other word processing program on the market.

.....

#### NEW ROUTINES

#### BATCHCOPY

Carl Singer sent me the following routine for batch file copies - the PIP type program I mentioned last month. I've been running it and it works great. Instructions are included in the source, but I'll repeat them. To run, type

```
BATCHCOPY n
      where "n" is the total number of files you
      want to copy.
```

BATCHCOPY is loaded and responds with an "\*\*\*". Then you type in the "n" files, just as you would if they were the parameters to a normal FILECOPY. Since BATCHCOPY is now resident in main memory, you can remove that disk and work on any set of disks, whether or not they contain BATCHCOPY. You can rename files and move files back and forth, all in the same command set.

After the "n" requests are entered, BATCHCOPY goes to work and you can leave your computer and do something important, like see what's going on with the TV football games.

For those of you who don't want to type in and assemble the source, BATCHCOPY is available on MUG Library Disk 9, in both source and executable versions.

```

0010 * PROGRAM $BATCHCOPY
0020 * By Carl J.Singer
0030 * 6049 N. Morgan St.
0040 * Alexandria, Va. 22312
0050 *
0060 * This is the source for BATCHCOPY,
0070 * a program that permits file copying
0080 * in batches. It should be a TYPE 18,
0090 * and is called as follows:
0100 *
0110 *     BATCHCOPY <number of files; maximum 50>

0120 *
0130 * Each copy will be prompted by an asterisk,
0140 * and is entered in the normal way. Example:

0150 *
0160 *     BATCHCOPY 5
0170 *     **"AMOS" "1:ANDY"
0180 *     **"LINEEDIT" 1
0190 *     **"1:UTILITY" "FUTILITY"
0200 *     **"2:BIG" "3:LITTLE"
0210 *     **"2:WHEATIES" 1
0220 *
0230 * After the last scheduled entry,
0240 * the routine will perform the
0250 * requested changes, in order.
0260 * The first disk error encountered
0270 * will abort the copy being worked
0280 * on and return to monitor.
0290 * All previous copies will be O.K.
0300 *
0310 *     LINK 'SYSQ1'
0320 *     LINK 'SYSQ2'
0330 *
0340 *     ORG @APROGRAM
0350 *
0360 *     LDA 4C9H
0370 *     CPI 40H
0380 *     JZ H2B3F
0390 *     LHLD @CONSOLEADDR
0400 *     LXI D,2
0410 *     DAD D
0420 *     MOV A,M
0430 *     INX H
0440 *     MOV H,M
0450 *     MOV L,A
0460 *     LXI D,H2B28
0470 *     LDAX D
0480 *     ORA A
0490 *     JZ @WARMSTART
0500 *     MOV B,A
0510 *     PUSH D
0520 *     PUSH H
0530 *     CALL H2B27
0540 *     POP H
0550 *     POP D
0560 *     INX D
0570 *     JMP H2B16
0580 *
0590 *     H2B27 PCHL
0600 *
0610 *     H2B28 DB 0AH,0DH
0620 *     DT 'SYSTEM VERSION ERROR'
0630 *     DB 0
0640 *
0650 *     H2B3F LDA @NBINPAR
0660 *     DCR A
0670 *     JNZ @DISKERROR-2
0680 *     LDA @BBUFF0
0690 *     ORA A
0700 *     JZ @DISKERROR-2
0710 *     MOV B,A
0720 *     ANI 0FH
0730 *     MOV C,A
0740 *     MOV A,B
0750 *     ANI 0F0H
0760 *     RRC
0770 *     RRC
0780 *     RRC
0790 *     RRC
0800 *     MOV B,A
0810 *     ADD A
0820 *     ADD A
0830 *     ADD B
0840 *     ADD A
0850 *     ADD C
0860 *     CPI 51
0870 *     JC $+5
0880 *     MVI A,50
0890 *     STA NUMFILES

```

```

0900 STA FILESLEFT
0910 LXI H,FILES
0920 SHLD FLPTR
0930 READIN CALL @CCRLF
0940 LXI H,PROMPT
0950 CALL @CILINE
0960 LXI H,@INBUFF
0970 CALL @PARAM
0980 JC @DISKERROR
0990 MVI C,23
1000 LHLD FLPTR
1010 LXI D,@NASC PAR
1020 CALL @TRANSDHC
1030 LXI D,@DRIVEN0
1040 MVI C,2
1050 CALL @TRANSDHC
1060 MVI C,2
1070 LXI D,@BBUFF0
1080 CALL @TRANSDHC
1090 SHLD FLPTR
1100 LDA FILESLEFT
1110 DCR A
1120 STA FILESLEFT
1130 JNZ READIN
1140 LXI H,FILES
1150 SHLD FLPTR
1160 LDA NUMFILES
1170 STA FILESLEFT
1180 *
1190 * GETPARAMS
1200 LHLD FLPTR
1210 XCHG
1220 LXI H,@NASC PAR
1230 MVI C,23
1240 CALL @TRANSDHC
1250 MVI C,2
1260 LXI H,@DRIVEN0
1270 CALL @TRANSDHC
1280 MVI C,2
1290 LXI H,@BBUFF0
1300 CALL @TRANSDHC
1310 XCHG
1320 SHLD FLPTR
1330 *
1340 * MOVEIT
1350 LDA @NASC PAR
1360 CPI 1
1370 JC @DISKERROR-2
1380 JNZ H2B5B
1390 MVI C,0AH
1400 LXI D,@ASCBUFF0
1410 LXI H,@ASCBUFF1
1420 CALL @TRANSDHC
1430 LDA @BBUFF0
1440 STA @DRIVEN1
1450 LXI D,H2C8F
1460 MOV H,D
1470 MVI L,OFFH
1480 MOV A,M
1490 INR M
1500 CMP M
1510 JZ H2B6C
1520 DCR M
1530 INR H
1540 JMP H2B61
1550 *
1560 * H2B6C
1570 DCR H
1580 XCHG
1590 CALL @DESUBHL
1600 MOV D,B
1610 MOV E,C
1620 LXI H,104H
1630 CALL @DEDIVHL
1640 MOV H,B
1650 MOV L,C
1660 SHLD H2C83
1670 MVI C,0
1680 CALL @TRANSFILENAME
1690 MVI B,0
1700 LDA @DRIVEN0
1710 MOV C,A
1720 LXI H,@FILEBUFFER0
1730 CALL @OPENFILE
1740 JC @DISKERROR
1750 CALL @RFILEINF
1760 JC @DISKERROR
1770 XCHG
1780 DCX H
1790 SHLD H2C85
1800 MVI C,1
1810 CALL @TRANSFILENAME
1820 MVI B,1
1830 LDA @DRIVEN1

```

```

1810      MOV    C,A
1820      LXI    D,0
1830      LXI    H,@FILEBUFFER1
1840      CALL   @CREATE
1850      JC     @DISKERROR
1860      XCHG
1870      SHLD   H2C87
1880      SHLD   H2C89
1890      SHLD   H2C8B
1900 H2BBE  LXI    H,H2C8F
1910      SHLD   H2C8D
1920 H2BC4  LHL    H2C87
1930      INX    H
1940      SHLD   H2C87
1950      LHL    H2C89
1960      INX    H
1970      SHLD   H2C89
1980      XCHG
1990      MVI    B,0
2000      CALL   1A8EH
2010      JC     @DISKERROR
2020      PUSH   H
2030      MVI    A,1EH
2040      CALL   1D6DH
2050      MOV    C,M
2060      INX    H
2070      MOV    B,M
2080      DCX    H
2090      INX    B
2100      INX    B
2110      XCHG
2120      LHL    H2C8D
2130      CALL   @TRANSDHBC
2140      POP    D
2150      MOV    M,E
2160      INX    H
2170      MOV    M,D
2180      INX    H
2190      SHLD   H2C8D
2200      LHL    H2C89
2210      XCHG
2220      LHL    H2C85
2230      CALL   @COMPARE
2240      JZ     H2C0E
2250      LHL    H2C83
2260      CALL   @DEMODHL
2270      MOV    A,B
2280      ORA    C
2290      JNZ   H2BC4
2300 H2C0E  LXI    H,H2C8F
2310      SHLD   H2C8D
2320 H2C14  MVI    A,4
2330      MVI    B,1
2340      CALL   1D6DH
2350      XCHG
2360      LHL    H2C8D
2370      MOV    A,M
2380      MOV    C,A
2390      STAX   D
2400      INX    H
2410      INX    D
2420      MOV    A,M
2430      MOV    B,A
2440      STAX   D
2450      INX    H
2460      XCHG
2470      MVI    A,1BH
2480      CALL   @HLADDA
2490      MOV    A,B
2500      ORA    C
2510      CNZ   @TRANSDHBC
2520      XCHG
2530      MOV    E,M
2540      INX    H
2550      MOV    D,M
2560      INX    H
2570      SHLD   H2C8D
2580      LHL    H2C8B
2590      INX    H
2600      SHLD   H2C8B
2610      XCHG
2620      MVI    B,1
2630      CALL   1AD3H
2640      JC     @DISKERROR
2650      LHL    H2C87
2660      DCX    H
2670      SHLD   H2C87
2680      MOV    A,H
2690      ORA    L
2700      JNZ   H2C14

```

```

2710      LHL    H2C89
2720      XCHG
2730      LHL    H2C85
2740      CALL   @COMPARE
2750      JNZ   H2BBE
2760      MVI    A,15H
2770      MVI    B,0
2780      CALL   1D6DH
2790      MOV    C,M
2800      MVI    A,15H
2810      MVI    B,1
2820      CALL   1D6DH
2830      MOV    M,C
2840      CALL   @CLOSEFILE
2850      JC     @DISKERROR
2860      MVI    B,0
2870      CALL   @CLOSEFILE
2880      JC     @DISKERROR
2890      LDA    FILESLEFT
2900      DCR    A
2910      JZ     @MDOSEXECUTIVE
2920      STA    FILESLEFT
2930      CALL   @CBRK
2940      JZ     @DISKERROR
2950      JMP    GETPARAMS
2960      *
2970 FILES  DS     1150
2980 FLPTR  DW     FILES
2990 NUMFILES DB    0
3000 FILESLEFT DB    0
3010 PROMPT DB    2AH,0DH
3020 H2C83  DW    0
3030 H2C85  DW    0
3040 H2C87  DW    0
3050 H2C89  DW    0
3060 H2C8B  DW    0
3070 H2C8D  DW    0
3080 H2C8F  DB    0
3090      *
3100      END

```

### CATALOG SYSTEM

by Buzz Rudow

I worked out a method for cataloging the contents of one's disks. It consists of the four programs described and listed below.

Several assumptions are made which put some restrictions on program operation:

- 1) You can only have 100 disks in a catalog.
- 2) No single disk can have more than 64 files, and those files must be in the front of the directory.
- 3) The average number of files per disk for your set on a catalog can not exceed 32.

Use of the catalog programs is done as follows. Load and run DIRINPUT. You have to start with disk 1, and then go in sequential order. You can never enter a disk number that exceeds "current + 1". If you have 6 disks input, you can then do number 7, but not 23 or 8 or 100.

Once a disk has been entered, it can then be re-entered. That is, suppose you enter 10 disks on Dec. 1. During December you change the contents of disks 5 and 7, and generate a new disk 11. On Jan. 1 you can "enter" all 3 of these disks. Disks 5 and 7 will write over the current information that the catalog has for them, and disk 11 will be appended on the end of the file.

You can list, either to the CRT or printer, the contents of the sequential entries by a load and run of DIRLIST.SEQ. You can generate a sorted version (by program name) of the sequential entries by a load and run of DIRSORT. Then you can list, either to the CRT or printer, the contents of the sorted entries by executing DIRLIST.SRT.

The program listings follow. If you don't want to type this all in, get Library Disk 10, which just has the catalog programs. They are expanded,

compared to the listings below, are menu driven, and include options for 3 catalogs, one of which is the implemented MUG Library. There is an additional version of DIRSORT which uses SORT/A, for those of you who have that. This catalog disk also contains the latest auto-configuration for the MUG. With a one-line change, which I can put in if I have your video terminal and its clear-screen code on file, all programs automatically determine MDOS Version (3 or 4) and proper clear screen action. More about auto-configuration next month.

In addition to typing in the following programs, you have to create the data files. From BASIC (don't use MDOS' CREATE), type -

```
OPEN 1 "N:DIRDAT.SEQ"
OPEN 2 "N:DIRDAT.SRT"
CLOSE 1
CLOSE 2
```

This set of programs does require a 2-drive system, though I imagine it could be changed to work on a single-drive.

```
10 ! DIRINPUT 11/19/81
20 DIM B$(7,250),W$(250),O$(1,3)
30 O$(0)=CHAR$(11): ! Clear Screen
40 J%=16R03B7
50 PRINT O$(0)
60 INPUT "Insert disk in drive 1, press RETURN to c
ontinue";A$
70 INPUT "ENTER DISK NUMBER";A$
80 IF LEN(A$)=0 PRINT "YOU MUST ENTER A NUMBER.":GOTO
TO 70
90 FOR I=1 TO LEN(A$)
100 IF MID$(A$,I,1)<"0" OR MID$(A$,I,1)>"9" THEN PR
INT "ALL CHARACTERS MUST BE NUMERIC.":GOTO 70
110 NEXT I
120 A=VAL(A$)
130 IF A=0 PRINT "ZERO IS NOT A LEGITIMATE DISK NUM
BER.":GOTO 70
140 IF A>100 THEN PRINT "THIS SYSTEM WILL ONLY SUPP
ORT 100 DISKS.":GOTO 70
150 OPEN 2 "DIRDAT.SEQ"
160 IF (A*4)-3<RECPUT(2) THEN GOTO 210
170 PRINT "YOU ONLY HAVE";SIZE(2)/4;"DISKS LOGGED."

180 PRINT "YOU CAN ADD DISK";(SIZE(2)/4)+1;"TO THE
END, BUT YOU CAN'T SKIP TO";A;"!"
190 CLOSE 2
200 GOTO 70
210 OPEN 1 "1:DIR"
220 EOF(1)=16
230 PUTSEEK(2)=(A*4)-3
240 FOR L%=3 TO 10 STEP 2: ! 8 sectors (64 files)
250 GET 1 RECORD L% A$
260 FOR K%=0 TO 7:B$(K%)="" :NEXT K%
270 GOSUB 400
280 WS=B$(0)+B$(1)+B$(2)+B$(3)+B$(4)+B$(5)+B$(6)+B$
(7)
290 GET 1 RECORD L%+1 A$
300 FOR K%=0 TO 7:B$(K%)="" :NEXT K%
310 GOSUB 400
320 PUT 2 WS;B$(0);B$(1);B$(2);B$(3);B$(4);B$(5);B$
(6);B$(7)
330 NEXT L%
340 EOF(1)=1
350 CLOSE 1
360 CLOSE 2
370 INPUT "DO YOU HAVE ANOTHER DISK";A$
380 IF LEFT$(A$,1)="Y" OR LEFT$(A$,1)="y" THEN GOTO
50
390 END
400 Q%=0
410 FOR K%=1 TO 128 STEP 16
420 IF PEEK(J%+K%)=255 THEN B$(Q%)=REPEAT$( " ",15):
GOTO 590
430 FOR I%=0 TO 9
440 B$(Q%)=B$(Q%)+CHAR$(PEEK(J%+(K%+I%)))
450 NEXT I%
460 P%=PEEK(J%+K%+11)
470 IF P%>127 THEN P%=8:GOTO 560
480 IF P%>27 THEN P%=7:GOTO 560
490 IF P%>23 THEN P%=6:GOTO 560
500 IF P%>19 THEN P%=5:GOTO 560
510 IF P%>15 THEN P%=4:GOTO 560
520 IF P%>11 THEN P%=3:GOTO 560
530 IF P%>7 THEN P%=2:GOTO 560
```

```
540 IF P%>3 THEN P%=1:GOTO 560
550 P%=0
560 B$(Q%)=B$(Q%)+FMT(P%,"9")
570 B$(Q%)=B$(Q%)+FMT(PEEK(J%+(K%+12)),"9")
580 B$(Q%)=B$(Q%)+FMT(PEEK(J%+(K%+13)),"999")
590 Q%=Q%+1:NEXT K%
600 RETURN

10 ! DIRLST.SEQ 11/15/81
20 DIM B$(3,250),O$(1,3)
30 O$(0)=CHAR$(11): ! Clear Screen
40 PRINT O$(0)
50 G%=0
60 PRINT "DO YOU WANT HARDCOPY"
70 INPUT B$
80 IF LEFT$(B$,1)="Y" THEN ASSIGN (2,3)
90 OPEN 1 "DIRDAT.SEQ"
100 X%=1
110 FOR P%=1 TO SIZE(1)/4
120 FOR L%=0 TO 3
130 GET 1 B$(L%)
140 NEXT L%
150 PRINT "DISK ";FMT(X%,"99")
160 GOSUB 330
170 PRINT "====="
180 GOSUB 330
190 FOR L%=0 TO 3
200 FOR K%=1 TO 240 STEP 15
210 IF MID$(B$(L%),K%,1)<>" " THEN PRINT MID$(B$(L%
),K%,15):GOSUB 330
220 NEXT K%
230 NEXT L%
240 X%=X%+1
250 PRINT
260 GOSUB 330
270 PRINT
280 GOSUB 330
290 NEXT P%
300 CLOSE 1
310 ASSIGN (2,2)
320 END
330 G%=G%+1
340 IF G%>=55 THEN FOR Z%=1 TO 11:PRINT:NEXT Z%:G%=
0
350 RETURN

10 !DIRSORT 11/15/81
20 SIZES (3,2,15)
30 DIM B$(3,250),S(60,2),L$(18,13),O$(1,3)
40 O$(0)=CHAR$(11): ! Clear Screen
50 OPEN 1 "DIRDAT.SEQ"
60 OPEN 2 "DIRDAT.SRT" CLEAR
70 DIM C$(8*SIZE(1),13)
80 Q%=1
90 FOR P%=1 TO SIZE(1)/4
100 FOR L%=0 TO 3
110 GET 1 B$(L%)
120 NEXT L%
130 FOR K%=0 TO 3
140 FOR K%=1 TO 240 STEP 15
150 IF MID$(B$(L%),K%,1)=" " OR MID$(B$(L%),K%,10)=
"DIR " THEN GOTO 180
160 C$(Q%)=MID$(B$(L%),K%,11)+FMT(INT((RECGET(1)+2)
/4),"99")
170 Q%=Q%+1
180 NEXT K%
190 NEXT L%
200 NEXT P%
210 GOSUB 515 : ! Sort
220 GOSUB 4115: ! Write data out
230 GOSUB 4215: ! Termination
240 CLOSE 1
250 ASSIGN (2,2):END
500 !
505 ! SORT SECTION
510 !
515 PRINT O$(0)
520 PRINT "STARTING SORT"
525 P=1
530 S(P,1)=1
535 S(P,2)=Q%
540 PRINT "PASS";X
545 X=X+1
550 IF P<0 GOTO 675
555 L=S(P,1)
560 H=S(P,2)
565 P=P-1
570 IF H<L GOTO 540
```

```

575 A=L
580 B=H
585 S=-1
590 IF A>=B GOTO 645
595 IF C$(A)<=C$(B) GOTO 620
600 F$=C$(A)
605 C$(A)=C$(B)
610 C$(B)=F$
615 S=-S
620 IF S<0 GOTO 635
625 B=B-1
630 GOTO 640
635 A=A+1
640 GOTO 590
645 IF A+1>=H GOTO 665
650 P=P+1
655 S(P,1)=A+1
660 S(P,2)=H
665 H=A-1
670 GOTO 570
675 PRINT "SORT COMPLETE FOR";Q$;"ENTRIES"
680 RETURN
4100 !
4105 ! Write data out
4110 !
4115 R=0
4120 FOR Q=0 TO Q%
4125 L$(R)=C$(Q)
4130 R=R+1
4135 IF R=19 GOSUB 4315
4140 NEXT Q
4145 RETURN
4200 !
4205 ! Termination of data out
4210 !
4215 IF R=0 GOTO 4240
4220 FOR I=R TO 18
4225 L$(I)=REPEAT$( " ",13)
4230 NEXT I
4235 GOSUB 4315
4240 CLOSE 2
4245 RETURN
4300 !
4305 ! Output data to disk
4310 !
4315 PUT 2 L$(0);L$(1);L$(2);L$(3);L$(4);L$(5);L$(6)
;L$(7);L$(8);L$(9);L$(10);L$(11);L$(12);L$(13);L$(
14);L$(15);L$(16);L$(17);L$(18)
4320 R=0
4325 RETURN

```

```

10 !DIRLST.SRT 11/15/81
20 SIZES (3,2,15)
30 DIM W$(250),O$(1,3)
40 O$(0)=CHAR$(11)
50 PRINT O$(0)
60 PRINT "DO YOU WANT HARDCOPY"
70 INPUT B$
80 IF LEFT$(B$,1)="Y" THEN ASSIGN (2,3)
90 GOSUB 6015: ! Read in data
100 PRINT O$(0)
110 PRINT " NAME TYPE DISK"
120 PRINT
130 FOR L%=0 TO W%
140 IF LEN(C$(L%))=0 OR LEFT$(C$(L%),1)=" " THEN GO
TO 280
150 A%=VAL(MID$(C$(L%),11,1))
155 ON A%+1 GOTO 160,170,180,190,200,210,220,230,24
0
160 A$="DAT":GOTO 250
170 A$="SRC":GOTO 250
180 A$="OBJ":GOTO 250
190 A$="OVL":GOTO 250
200 A$="BAS":GOTO 250
210 A$="SYS":GOTO 250
220 A$="USR":GOTO 250
230 A$="RWD":GOTO 250
240 A$="UDF"
250 PRINT LEFT$(C$(L%),10);" ";A$;" ";RIGHT$(C$(
L%),2)
260 G%=G%+1
270 IF B$="Y" AND G%=55 THEN FOR H%=1 TO 9:PRINT:NE
XT H$:PRINT " NAME TYPE DISK":PRINT:G%=0
280 NEXT L%
290 ASSIGN (2,2):END
6000 !
6005 ! Read in data
6010 !
6015 OPEN 2 "DIRDAT.SRT"

```

```

6020 W%=SIZE(2)*19
6025 DIM C$(W%,13)
6030 K%=0
6035 FOR Q%=1 TO SIZE(2)
6040 GET 2 W$
6045 FOR J%=0 TO 18
6050 C$(K%+J%)=MID$(W$,(J%*13)+1,13)
6055 NEXT J%
6060 K%=K%+19
6065 NEXT Q%
6070 CLOSE 2
6075 RETURN

```

.....

#### ANOTHER CDS SOURCE

To help you CDSers with your repair problems, the House of Computers, Inc., is another dealer who has parts and repair facilities. Actually, Mark Herzog, President of House of Computers, says he has full new systems for sale. They aren't called CDS, but keep the Versatile name. Custom software is also available. Hope to elaborate on Mark's products in a Micropolis Based Systems article in the near future. Contact Mark at 368 Eglinton Ave. West, Toronto, Ontario M5N 1A2 Canada, (416) 482-4336.

.....

#### LETTERS

##### KILOBAUD

Dear Mr. Rudow:

Thank you for your letter of some months back regarding your desire to convert Kilobaud programs for distribution to Micropolis users. My apologies for taking so long to answer your inquiry.

We have no objection to your proposed use of our material, provided the converted programs are not sold for profit and provided they carry a comment or remark line giving Kilobaud Microcomputing as the source for the original program. We also expect the program author to receive a credit line.

Thanks for your interest in our publication.

Jeffrey D. DeTray, Editorial Manager, Kilobaud Microcomputing, Peterborough, NH 03458

##### PROGRAMS FROM MAGAZINES

ED. NOTE: We can start a new disk anytime you people want to. Start looking back through your Kilobaud & BYTE issues for applicable software. This disk will be distributed the same as Disk-6, the Micropolis disassemblies. It will not be available for \$15. The only way to get it is to submit a program on your disk, and \$3. Please specify that your program is from one of the magazines - both in a note and in the code.

.....

##### DISK FORMAT CONVERSION

Buzz,

I have both 8-inch and Micropolis Mod II (5.25 inch) disk drives. This allows me to convert the CP/M user group library programs to Micropolis Mod II format. I would be willing to provide this conversion service to any interested MUG members at a nominal fee. Please let me know if there is an interest in this kind of thing.

Charles L. Pfau  
6223 South 72nd Ave., Ralston, NE 68127

.....

##### SORCERER INFORMATION

Buzz,

You may remember that I asked you for help some time ago regarding problems I was having with my

Sorcerer. Well, I have everything resolved thanks to some great people in Sydney, the Software Source, P.O. Box 364, Edgecliff, N.S.W., Australia, 2027.

For the Sorcerer owners back in the states, you might mention that the Software Source is offering a fully configured version of Lifeboats CP/M 2.2 which includes some super operating enhancements tailored to the Sorcerer hardware while retaining complete Lifeboat compatibility.

Daniel Jamba, P.O. Box 2477  
Alice Springs, Australia 5750  
.....

CLASSIFIED

WANTED: To correspond with anyone interested in interfacing Mp to the 16-bit systems, especially the Godbout 8085/8088 board. Also would like to establish a Mp subgroup in the NY area.

Walter Garrett, 212/595-2835  
138 W. 82nd St., NY NY 10024  
.....

WANTED: I'd like to interface my 1053-II's to an 8086 system on the S-100 Bus (Seattle Computer Products 2-card set). Anybody doing such non-standard stuff?

Richard J. Hanney, 805/985-8308  
C/O Special Systems Technology, Inc.  
3533 Old Conejo Rd.,  
Suite 111, Newbury Park, CA 91320  
.....

FOR SALE: Micropolis MOD I disk unit, controller, PROMs to relocate at 2000K (for POLY-88), manuals, and 2 boxes of disks. \$500.

Dr. Donald O. Inslee,  
2001 Castlebridge Rd., Midlothian VA 23113  
.....

WANTED: Vector Mindless Terminal, Flashwriter board and Documentation.

Dr. Kevin Geyer, (714) 962-8240  
8591 Whitesails Cir., Huntington Beach CA 92646  
.....

WANTED: Any information available on Computer Aided Instruction (CAI) for the Micropolis system. Using computer in grade school.

Mauricio Gluck, (305) 673-0866  
4510 Pinetree Dr., Miami Beach FL 33140  
.....

WANTED: Any programs running on the Micropolis system which aid in the management and accounting of stocks, bonds, and other investments.

John G. Scott, (919) 275-2881  
E. F. Hutton & Co., Inc.  
P.O. Box 749, Greensboro NC 27402  
.....

WANTED: A D.C. Hayes or other self-contained modem.

Robert S. Hoover, (714) 724-1513  
1875 Monte Vista Dr., Vista CA 92083  
.....  
12/01/81

WANTED: The current address or phone number of Michael Shroyer. Or, can anyone help me patch Electric Pencil to run on 56K VG. Have 48K EP.

Hans Rakow, (403) 467-8630, home  
(403) 436-4414, work  
34 Wren Crescent, Sherwood Park, Alta. CANADA  
T8A 0G5

Published Monthly by the MUG  
Subscription rates:  
U.S., Canada, Mexico; \$18/year; Other, \$25/year

FIRST CLASS MAIL  
=====

FIRST CLASS MAIL  
=====

MICROPOLIS USERS GROUP

Buzz Rudow, Editor  
604 Springwood Circle  
Huntsville AL 35803  
(205) 883-2621

FIRST CLASS MAIL  
=====