

MICROPOLIS USERS GROUP

MUG Newsletter # 16 - November 1981

WRITING GOOD PROGRAMS (PART 4)by Burks A. Smith of DATASMITH
Box 8036, Shawnee Mission, KS 66208

Most programs are filled with loops, or parts of a program that repeat. After all, one of the things a computer does best is doing something over and over again. Loops serve many purposes, but there are only a few different kinds of loops that can be built in BASIC, so it is best to define what they are. By breaking down loops into simple building blocks, the programmer can gain a clearer idea of how best to use them without introducing bugs into a program.

In their simplest form, loops have three parts: An entry point, a "body" where the processing is done, and an exit point. Except for loops that repeat forever (not very popular), every loop also has some sort of test built in to determine when looping stops and some other process begins. The two most common places for this test are at the beginning or at the end of the body, and by determining where the test is, loops structures can be divided into two major types:

Loops that have a test at the end of the body are called REPEAT-IF loops. This type of loop is characterized by the fact that the body of the loop is always executed at least once when the loop is entered. Being at the end, the test determines whether the body should be executed again. A FOR-NEXT loop in Micropolis BASIC is a REPEAT-IF structure, in that the test occurs when the NEXT statement is encountered. For this reason, a FOR-NEXT loop will always do one iteration even though the FOR value may exceed the TO value when the loop is entered. The statement:

FOR X=99 TO 1

will let execution continue to the NEXT X statement even though 99 is greater than 1 at the outset. Programmers should be aware of this characteristic, to avoid trouble with loops that may use variables to set the starting and ending points of a FOR-NEXT loop.

It is interesting to note at this point that not all BASICs use a REPEAT-IF structure for FOR-NEXT loops. Some test at the beginning and will not execute the body of the loop at all if the starting point exceeds the ending point at the outset. This would seem preferable, at least to me, but some of these pretest type FOR-NEXT loops do have a rather annoying side effect: They exit with the control variable set at one greater than the TO value, if counting in a positive direction, and one less than the TO value if counting negatively. Micropolis FOR-NEXT loops always end with the control value exactly equaling the TO value. I am told that the Micropolis (REPEAT-IF) structure for a FOR-NEXT loop is not ANSI standard, and the other type is. Actually, I don't think you can get a version of BASIC that is totally ANSI "standard", so the important thing is that you know the characteristics of your language. It is also important that the language be predictable in its behavior or all bets are off. I recently read an article documenting bugs in some "home-computer" versions of BASIC that cause all sorts of strange things to happen when FOR-NEXT loops were not exited "normally" through the NEXT statement. As far as I know, Micropolis FOR-NEXT loops are predictable in their behavior and are without any bugs.

Loops that have a test at the beginning of the body are called WHILE-DO loops. As already noted, this structure prevents the loop even being entered if the condition does not exist, and will continue looping as long as the WHILE condition is met.

Many computer languages, including some BASICS, have some kind of WHILE statement built in, but the structure is quite easy to implement without a special command. Simply begin the loop with an IF statement that skips the body if the WHILE condition is not met. At the end of the body, GOTO the IF statement at the beginning and test again. This takes exactly the same number of lines as it would if a WHILE-WEND statement were built into the language.

Adding variations to the simple types of loops above, or by combining them will yield a variety of other structures. As a programmer, it is important to recognize the type of loop you will need for your application, then design the loop in a manner that produces predictable results. One of the most common types of bugs in programs is a loop that does not perform according to specifications, because it was designed in a sloppy manner.

This month concludes the "writing good programs" series, and I hope my thoughts have been useful to those desiring to improve their proficiency at programming. In future columns, I intend to concentrate on documenting some of the "undocumented" features of Micropolis BASIC and MDOS, as well as expanding on the material already in the manual. If you have any suggestions on what you would like to see here, or if you have any questions you would like answered, please write.

.....

BASIC PROGRAMMING TIPSby Gene Riding
2227 Chicago St., San Diego CA 92110

SIZE(n)

Here is a routine using the SIZE(n) Statement you might want to use in your programs to reduce disk search time. In the payroll program I use in my business, I put from 4 to 6 entries (each being a record) in the file each week. The data I need for the Payroll Report each week is searched by date. As the file gets longer the search time becomes intolerable.

After trying a number of schemes to move the GET pointer, I realized the simplest method was to use the SIZE statement. Since the data I am searching for is always at the end of the file, I used the following routine, written here in dummy form.

```
100 INPUT "DATA";D$
110 OPEN 1 "FILENAME"
120 FOR X=SIZE(1)-6 TO SIZE(1)
130 ! GET DATA FROM RECORD
140 ! COMPARE THE DATA FOR PROPER DATE
150 ! IF INPUT DATE = DATE IN RECORD
160 ! THEN CALL PRINT ROUTINE
170 NEXT X
```

Instead of searching the entire file, now I only search the week of interest. A week in December takes no longer than a week in January.

.....

MOTIONby Anthony Pickert
5623 Sloan, Kansas City KS 66104

The concept of motion on the screen is simple if one were to realize that illusions do exist. The film-making of the early 1900's proved this. The same concept is shown here by the act of showing a character, then blanking the same position before showing the character in the next position. The smaller the increment, the more fluid the motion appears. The early films used large increments making the picture appear choppy and fast.

The value of such ability on the computer is, at first, opaque. We have not fully applied technol-

ogy to such areas as education and communication. The state of the computer industry is ever changing, though, and one is thus encouraged to learn concepts rather than peculiar language. The value of this program is in its concept, as well as in its documentation of the performance of hardware and software alike.

The particular chip is Z-80A Mostek, at 4 megahertz speed. The video baud rate is 9600, and language is Micropolis. The round trip speed of the program run, for 80 columns wide, is 5.7 seconds. To test for efficient programming try to vary the way in which the starting screen address is added or set. For example: F=16RF000 + 16R190. (1*16*16 + 9*16=400).

One final note: To test the keyboard for any response during the program execution the function K=IN(1) could be set inside the on/off loop. The ASCII value 52 will be given for pressing the number 4 on your keyboard. Insert such test instructions at line 190 to delay the showing of the character and not the hiding of it. Thus, with more programming logic, you could control the motion of the character from the keyboard.

```

100 I M O T I O N
110 PRINT CHAR$(4);CHAR$(14);
120 F=16RF000 +400
130 B1=0:B2=79:S=1
140 FOR B=B1 TO B2 STEP S
150 C=27:Z=0
160 POKE(F+B)=C
170 IF Z=1 THEN 190
180 Z=1:C=32: GOTO 160
190 NEXT B
195 IF S=-1 THEN 130
200 B1=B:B2=0:S=-1
210 GOTO 140
220 END
300 ! Round trip = 5.7 seconds
310 ! 110 = Clear screen and hide cursor
320 ! 120 = Screen address and offset
330 ! 130 = begin;end;step of loop at 140
340 ! 140 = Definition of a loop at 190
350 ! 150 = Character and flag (on/blank)
360 ! 160 = Poke on screen the character
370 ! 170 = Test for a flag (on)
380 ! 180 = If 'on' before, then blank now
390 ! 190 = end of one direction (or loop)
395 ! 195 = If backward before then forward now
400 ! 200 = Otherwise go backward with -1 step
410 ! 210 = Transfer logic to line 140
420 ! 220 = Physical end of this program

```

ED NOTE: This is kind of cute. I tried it on my SOL. You will need to change line 110 to have your own clear screen command (for the CHAR\$(4)). If you can't hide the cursor (the CHAR\$(14)), don't worry about it. The program works fine even if the cursor shows. This program will only work for memory mapped systems. You need to change the 16RF000 at line 120 to be the start of your video memory. The '400' is a 5-line offset for an 80-column screen. Change it to '320' for a 64-column screen. The B2=79 in line 130 should also be changed to B2=63 to get the same effect. The setting of C=27 in line 150 defines the character to be displayed. If '27', which is the special character ESCAPE, doesn't display, substitute anything you'd like.

After "watching the bouncing ball" scoot around for awhile, I wondered how fast it would move if the program was compiled under BASIC/S. The answer is much faster - so fast that the screen update doesn't keep up.

.....

TURNING OFF DRIVES BETWEEN ACCESSES

Several people have sent me schematics of modifications they implemented on their older drives, to have them turn the motors off like the newer drives. I have received some half dozen ways to produce the same end result.

Rather than publish any of them, I'll refer you to the source. Micropolis has a product Bulletin #FD-0014 which shows just how to make the modification to any model of controller or drive printed circuit board. Write Tim Matthews and request a copy, if you're interested. (Micropolis Corp., 21329 Nordhoff St., Chatsworth CA 91311, 213/709-3300)

The motors in the older drives run continuously if a disk is inserted and latched. More recent drives have the motor run when the disk is first latched (to position it), and then only when the drive is accessed. The newer method saves wear on the motors and disks, and reduces heat.

I've heard a rumor that there is some problems with the compatibility of Lifeboat's CP/M and the modification. If the drive is off when data is requested, a few milliseconds must be taken to get the disk up to speed. CP/M supposedly doesn't allow the time. Again, this is just a rumor. If anyone has CP/M running on a "new" drive, let me know how it works.

Is the modification necessary? Well, I've had an "old" double-MOD II for three years and never had a failure. I do have a habit of unlatching the disk if I'm not going to need it for awhile, however. This habit causes occasional "DISK OFFLINE" failures when I try to read from the disk and haven't relatched it. Life would be easier if the mod was put in. If you are doing communications and leave your system up all night, I'd say the mod is very necessary.

.....

SOLVING THE JUMBLES

by Gene Riding
2227 Chicago St., San Diego CA 92110

The JUMBLE puzzle appears in many newspapers across the country. Since the puzzle is computer generated, why not solve it with the same tool. Listed below is a program which can do it.

```

100 REM **JUMBLE SOLVER**
105 DIM O$(0,4)
106 !
107 ! THE FOLLOWING DIRECTIVE SETS THE CLEAR SCREEN
    COMMAND
108 ! EDIT LINE 110 FOR YOUR COMPUTER
109 !
110 O$(0)=CHAR$(4): ! FOR VECTOR MINDLESS TERMINAL
111 !
115 DIM L$(6,6)
120 DIM M$(6,6)
130 DIM P$(6,6)
140 PRINT O$(0)
150 PRINT "THIS IS A WORD UNSCRAMBLER PROGRAM."
160 PRINT "YOU MAY INPUT ANY 4, 5, OR 6 LETTERS AND"
170 PRINT "I WILL PRINT ALL POSSIBLE COMBINATIONS OF"
180 PRINT "THOSE LETTERS."
190 PRINT
200 PRINT "TO INPUT YOUR WORD LENGTH TYPE 4,5,OR 6";
210 INPUT N
220 IF N=4 THEN 250
230 IF N=5 THEN 430
240 IF N=6 THEN 630
250 PRINT
260 PRINT"TYPE ANY FOUR LETTERS SEPARATED BY COMMAS"
270 INPUT L$(1),L$(2),L$(3),L$(4)
280 PRINT
290 FOR A1=1 TO 4
300 FOR A2=1 TO 4
310 IF A2=A1 THEN 380
320 FOR A3=1 TO 4
330 IF A3=A1 THEN 370
340 IF A3=A2 THEN 370
350 LET A4=10-(A1+A2+A3)
360 PRINT L$(A1);L$(A2);L$(A3);L$(A4),
370 NEXT A3
380 NEXT A2
390 NEXT A1
400 PRINT:PRINT
410 INPUT "AGAIN";Y$:IF LEFT$(Y$,1)="Y" GOTO 140
420 END

```

```

430 PRINT
440 PRINT"TYPE ANY FIVE LETTERS SEPARATED BY COMMAS"
450 INPUT M$(1),M$(2),M$(3),M$(4),M$(5)
460 FOR B1=1 TO 5
470 FOR B2=1 TO 5
480 IF B2=B1 THEN 600
490 FOR B3=1TO 5
500 IF B3= B1 THEN 590
510 IF B3=B2 THEN 590
520 FOR B4 =1 TO 5
530 IF B4 =B1 THEN 580
540 IF B4 =B2 THEN 580
550 IF B4 =B3 THEN 580
560 LET B5=15-(B1+B2+B3+B4)
570 PRINT M$(B1);M$(B2);M$(B3);M$(B4);M$(B5),
580 NEXT B4
590 NEXT B3
600 NEXT B2
610 NEXT B1
620 GOTO 400
630 PRINT
640 PRINT"TYPE ANY SIX LETTERS SEPARATED BY COMMAS:"
650 INPUT P$(1),P$(2),P$(3),P$(4),P$(5),P$(6)
660 PRINT
670 FOR C1=1 TO 6
680 FOR C2=1 TO 6
690 IF C1=C2 THEN 870
700 FOR C3=1 TO 6
710 IF C3=C1 THEN 860
720 IF C3=C2 THEN 860
730 FOR C4=1TO 6
740 IF C4=C1 THEN 850
750 IF C4=C2 THEN 850
760 IF C4=C3 THEN 850
770 FOR C5= 1TO 6
780 IF C5=C1 THEN 840
790 IF C5=C2 THEN 840
800 IF C5=C3 THEN 840
810 IF C5=C4 THEN 840
820 LET C6=21-(C1+C2+C3+C4+C5)
830 PRINT P$(C1);P$(C2);P$(C3);P$(C4);P$(C5);P$(C6),
840 NEXT C5
850 NEXT C4
860 NEXT C3
870 NEXT C2
880 NEXT C1
890 GOTO 400
900 END
    
```

.....

CURRENT MUG PRICE LIST

Product	US, Canada and Mexico	Airmailed Elsewhere
-----	-----	-----
Newsletter, 12 months: (Surface mail)	\$18 ---	\$25 \$20
Newsletter, 3 month trial	\$ 5	\$ 7
Newsletter, single issue	\$ 2	\$ 3
Newsletter, back issues (12)	\$12	\$25
Library Disks, #1-5, #7 With disk and program or article -	\$ 3*	\$ 5*
Without prog/article - MOD II/MOD I	\$15/20	\$17/22
Library Disk, #6 With disk and program or article -	\$ 3*	\$ 5*
Without prog/article - Everyone must submit Ver. 4 RES & MDOS modules with request for disk 6.	Not Available	
Catalog of Library contents and back issue index	\$ 1	\$ 2

* MOD I must submit 2 disks for every library disk desired. All 'trades' must include a program and a disk for each library disk. Otherwise, I'll supply the extra disk for \$5.

Disks 1, 2 and 7 are application and utility programs. Disk 3 contains games. Disk 4 contains MUG membership file and application programs. Disk 5 contains S/W Vendors information and appli-

cation programs. Disk 6 contains disassemblies of RES, MDOS, and LINEEDIT and other system information.

VISA and MASTERCARD accepted. All prices are "cash" (check or moneyorder in U.S. funds). Charge orders billed at 104% of listed prices.

.....

AVAILABLE LIBRARY DISK

Those joining this year haven't seen the contents of the library disks 1 - 3. Disks 4 - 6 haven't been detailed at all. Disk 7 is new this month. Listed below are the details of all current MUG library disks.

THE MUG LIBRARY

MUG MOD II Library Disk 01, Revsion 00

Name	Type	Rev	Author/Description
====	====	===	=====
MONEYONE	BAS	00	Hogan, T. Load and run for control of the following 4 programs for home money management.
LOANS	BAS	00	
NETWORTH	BAS	00	
CHECKBOOK	BAS	00	
BUDGET	BAS	00	
GRADEBOOK	BAS	00	Hogan, T. Run in conjunction with the following program.
REPORTCARD	BAS	00	
AMORT	BAS	00	Smith, B. Amortization, with its description in the following file.
AMORT-TEXT	SRC	00	
POSTMAN2.0	BAS	00	Hogan, T. Mail-list routine.
DEPREC	BAS	00	Shapiro, J. Depreciation.
AMORT.S	BAS	00	Shapiro, J. Amortization.
INVOICE	BAS	00	Shapiro, J. Invoice generation.
FORMULA1	BAS	00	Shapiro, J. This and the following program are variable rate mortgage calculators.
FORMULA2	BAS	00	
STATISTICS	BAS	00	Hogan, T. Control program for the following 5 statistical programs.
MMS-MEAN	BAS	00	
MMS-LINCOR	BAS	00	
MMS-T-TEST	BAS	00	
MMS-ANOVA	BAS	00	
MMS-COVAR	BAS	00	
VISLETTER	BAS	00	Anders, M. A nice example of the rudiments of a BASIC word-processor.
LOADGO	SRC	00	Micropolis. Allows automatic execution of BASIC application program from initial boot.
POKE.VIDEO	BAS	00	Maschino, G. Some elements of graphics manipulation for a memory mapped video in this and the next program.
POKE.NRS	BAS	00	
DATES	BAS	00	O'Brien, D. Converts a numeric date to day of week.
RLTIME.DOC	SRC	00	Rowland, H. Real-time clock routines description, with assembly language source in next file.
RLTIME.SRC	SRC	00	
START	BAS	00	Anders, M. Load and run for control of the following 9 game routines.
DATA	DAT	00	
KEY	BAS	00	
CRAZYTALK	BAS	00	
BATTLESHIP	BAS	00	
LUNARLAND	BAS	00	
NERVES	BAS	00	
SHUTTLE	BAS	00	
IGUESS	BAS	00	
YOUNGUESS	BAS	00	
SPACETAXI	BAS	00	

SPACCAPTUR	BAS	00	Maschino, G. Game - Capture the Klingon.
BIGNUM	BAS	00	Rudow, B. Game which converts numeric input to text.
TICTACTOE	BAS	00	Smith, B. Game - with the following file being description.
TICTACTEXT	SRC	00	
ROMAN	BAS	00	Mac Kenzie, M. Converts arabic to roman numerals.
ONT-PAY	BAS	00	Mac Kenzie, M. A Canadian payroll program.
GENSORT	BAS	00	Burkhardt, E. Merges a set of unsorted records into a sorted file.
TRANS	BAS	00	Rudow, B. A BASIC filecopy utility.
BARCOM	SRC	00	Barnum, B. Assembly language Communications routine.
CALENDAR	BAS	00	Mac Kenzie, M. Generates a calendar.
NAMESORT-1	BAS	00	Mac Kenzie, M. BASIC bubble sort of names via DATA statements.
NAMESORT-2	BAS	00	Mac Kenzie, M. Same as above, except names via Keyboard input.
POKE.1	BAS	00	Maschino, G. Combination of sorts and graphics for memory mapped video in this and the following two programs.
POKE.2	BAS	00	
POKE.3	BAS	00	

MUG MOD II Library Disk 02, Revision 00

Name	Type	Rev	Author/Description
====	====	===	=====
TIMECARDS	BAS	00	Marion, H. Generates time-cards for either 2 or 4 entries per day.
REGTICARD2	BAS	00	
REGTICARD4	BAS	00	
TAPEREC	BAS	00	Shapiro, J. Aids in generating a balanced tape when taping records- see newsletter 11.
TAPE.TEXT	SRC	00	
P/DIM	BAS	00	Zale, B. BASIC save memory, restore memory routines - see newsletter 10.
RETRIEVE	BAS	00	
INKEY.B	BAS	00	Rudow, B. Substitute for the Microsoft INKEY function - see newsletter 9.
INKEY.T	SRC	00	
INKEY.L	OBJ	00	
DATAFILE	BAS	00	Hogan, T. Database manager for personal data.
BPOLE	BAS	00	Callaway, J. Tests screen or printer with characters - see newsletter 6.
STATEABB	BAS	00	Shapiro, J. Lists state abbreviations.
ASMBASIC	BAS	00	Rudow, B. Assembly language routines as shown in nsl'r 5.
ASMTEXT	OBJ	00	
PRINT.DIR	BAS	00	Maschino, G. Automatically sequences through and prints the DIR files of your disks.
ENTRYV3	BAS	00	Warner, D. Involved VISICALC type of program. Makes extensive use of SOROC Ctl Codes.
GRADES	BAS	00	Raney, M. Set of prorams for the teacher which record, weight, and average test scores.
GRAD.TEXT	SRC	00	
LOG	DAT	00	
READING1	DAT	00	
READING2	DAT	00	
READING3	DAT	00	
WRITING1	DAT	00	
WRITING2	DAT	00	
WRITING3	DAT	00	
MATH1	DAT	00	
MATH2	DAT	00	
MATH3	DAT	00	
GRAPHIC-B	BAS	00	Smith, B. Graphics program for Vector Graphic, followed by a fast assembly language version - see newsletter 11.
S/PLOT	SRC	00	
P.M.W.P.	BAS	00	MacKenzie, M. The Poor Man's Word Processor. Uses BASIC DATA statements.
S/SORT	BAS	00	Findlay, K. Set of programs for automatic set-up and execution of Systemation's SORT/A
SORT	BAS	00	
SYSDISK	DAT	00	
SORTOUT	DAT	00	- SORTDOC is documentation.

LISTDAT	BAS	00	
CHAINTEST	BAS	00	
SORTDOC	SRC	00	
S59000	BAS	00	Rudow, B. Four routines forming basis for auto-configuration. Not implemented. S60000
S60000	BAS	00	
S61000	BAS	00	
S57000	BAS	00	is an expanded and modified version of Dave Land's configuration in newsletter 8.
VERIFY	BAS	00	Callaway, J. Tells user if a variable is all blanks or zero length.
FILTER	BAS	00	Takes string, purges \$, comma, and converts to number - or issues an error message.
FINDSPACE	BAS	00	Finds first char in last name.
DATEFORM	BAS	00	Formats a date entered in numeric form.
TITLES	BAS	00	MacKenzie, M. Centers and boarders titles.
COPY	BAS	00	Sugarman, D. A BASIC filecopy program.
RENUM	BAS	00	Renumbers a file that has left most chars as record number.
REPORTER	BAS	00	Special formatting print pgm.
REPORT	BAS	00	
LOOK	BAS	00	Callaway, J. Allows inspection of files.

MUG MOD II Library Disk 03, Revision 00

Name	Type	Rev	Author/Description
====	====	===	=====
ATOM-20	BAS	00	Land, D. Games, Games, Games.
LUNAR	BAS	00	Nothing guaranteed at the moment. However, most look like they contain no control characters.
SORCERY	BAS	00	
ELIZA	BAS	00	
STARTREK	BAS	00	
SPACEWARS	BAS	00	
RSP	BAS	00	
CIVILWAR	BAS	00	
KIDMATH	BAS	00	
LOVE	BAS	00	
8-BALL	BAS	00	
ANIMALS	BAS	00	
SLOT	BAS	00	
CHECKERS	BAS	00	
CHECKERS1	BAS	00	
GAME	BAS	00	
GUESS	BAS	00	
TICTACTOE	BAS	00	
CHARACTERS	OBJ	00	
BANNER	BAS	00	
PIE	BAS	00	
PIEPAN	DAT	00	
BOAT	DAT	00	
NEWBANNER	BAS	00	
ADVENTURE	BAS	00	Land, S. And more games. Including the famous ADVENTURE!!
ADVP2	BAS	00	
AMESSAGE	DAT	00	
LIST	DAT	00	
ATEMS	DAT	00	
LIFE	BAS	00	
CAMEL	BAS	00	
TICTACTOE2	BAS	00	Callaway, J. The third implementation of TICTACTOE, all different.
BATTLESHIP	BAS	00	
MATH	BAS	00	
HI-LO	BAS	00	
ACEYDUCEY	BAS	00	Rudow, T. Not to be outdone, my daughter also has a few games in the system.

MUG MOD II Library Disk 04, Revision 16

Name	Typ	Rv	Sze	Author/Description
====	====	===	===	=====
MENU	BAS	01	00D	Rudow, B. This is the membership system.
MUGPRNT.S	BAS	00	010	Rudow, B. Print data by Seq.
MUGPRNT.N	BAS	00	008	Rudow, B. Print data by name.
MUGPRNT.M	BAS	00	008	Rudow, B. Print data by manu
MUGPRNT.P	BAS	00	008	Rudow, B. Print data by perf
MUGPRNT.Z	BAS	00	008	Rudow, B. Print data by ZIP
MUGLAB.S	BAS	00	007	Rudow, B. Print labels by Seq
MUGLAB.Z	BAS	00	008	Rudow, B. Print labels by ZIP
MUGLAB.ZS	BAS	00	009	Rudow, B. Print selected

				labels by ZIP
MEMBERS	DAT	16	104	Rudow, B. Member file
MEMBER.N	DAT	16	007	Rudow, B. Pointers for name
MEMBER.M	DAT	16	007	Rudow, B. Pointers for manuf
MEMBER.P	DAT	16	007	Rudow, B. Pointers for perif
MEMBER.Z	DAT	16	007	Rudow, B. Pointers for ZIP

MUG MOD II Library Disk 05, Revision 00

Name	Typ	Rv	Size	Author/Description
====	===	==	===	=====
MENU	BAS	00	00C	Rudow, B. This is the S/W Vendor system. See News'r #11
VENPRT	BAS	00	00E	Rudow, B. Print seq vendors
S/WPRT	BAS	00	012	Rudow, B. Print seq S/W
VENPRT.N	BAS	00	008	Rudow, B. Print alpha vendors
S/WPRT.T	BAS	00	00E	Rudow, B. Print S/W by type
S/WPRT.N	BAS	00	00E	Rudow, B. Print S/W by Vendor, short
S/WPRT.F	BAS	00	00E	Rudow, B. Print S/W by Vendor, long
S/WPRT.P	BAS	00	008	Rudow, B. Print S/W by Program name
VENDOR	DAT	00	05F	Rudow, B. Vendor file
S/W-LIST	DAT	00	12B	Rudow, B. S/W file
VENDOR.S	DAT	00	003	Rudow, B. Pointers for alpha vendors
S/W.T	DAT	00	008	Rudow, B. Pointers for S/W by type
S/W.N	DAT	00	008	Rudow, B. Pointers for S/W by vendor name
S/W.P	DAT	00	008	Rudow, B. Pointers for S/W by program name

MUG MOD II Library Disk 06, Revision 05

Name	Typ	Rv	Size	Author/Description
====	===	==	===	=====
EDITTEXT	SRC	00	013	Manderson, R. Documentation for the following 9 files.
LINEEDIT5	SYS	02	00F	Manderson, R. The assembled version of the EDIT files. Similar to Mp LINEEDIT, but has MERGE, enhanced APPEND, & editing capabilities.
SYSQ1	SRC	00	00B	Manderson, R.
EDIT1	SRC	00	00B	Manderson, R.
EDIT2	SRC	00	021	Manderson, R.
EDIT3	SRC	02	027	Manderson, R.
EDIT4	SRC	00	016	Manderson, R.
EDIT5	SRC	00	01B	Manderson, R.
EDIT6	SRC	00	013	Manderson, R.
DISYMS	SRC	00	021	Manderson, R. Instructions for assembly & use, as well as the source file, for a Z80 disassembler.
UNASSM	SYS	00	00B	Manderson, R. The assembled version of DISYMS. Lots of options- lists, ASCII equivalents, SYM tables.
MDOSTEXT	TXT	00	00C	Manderson, R. This and the next file have documentation for the 9 files which follow. Loads of info on the operation of RES and MDOS, both here & in the comments in the source code modules.
MDOSDOC	TXT	00	032	Manderson, R.
LOADER	SRC	00	00C	Manderson, R.
RESIO	SRC	00	020	Manderson, R.
RES1	SRC	00	036	Manderson, R.
RES2	SRC	00	034	Manderson, R.
RES3	SRC	00	020	Manderson, R.
MDOS1	SRC	00	03A	Manderson, R.
MDOS2	SRC	00	037	Manderson, R.
MDOS3	SRC	00	035	Manderson, R.
MDOS4	SRC	00	023	Manderson, R.
S/IODOC	TXT	00	00F	Callaway, J. Documentation for S/IORES, S/M, & VECTOR RES.
VECTOR RES	SRC	00	030	Callaway, J. For Ver. 4 of Vector Graphic Sys B. Contains true RUBOUT (Replaces with blank). Contains an auto load which automatically runs S/M file while in MDOS. Also

S/M	SRC	00	003	autoloads for BASIC. Callaway, J. The autoload program for VECTOR RES.
S/IORES	SRC	00	030	Callaway, J. For ALTAIR 8800B Contains S/W driver for 3port parallel Diablo Hytype I.
MILLER	TXT	00	00B	Miller, W. Documentation for MILLRES.
MILLRES	SRC	00	016	Miller, W. For a VDB 8024 Video Card and Dataproducts 2310 printer. Contains CONTROL-P trap for switching output from screen to printer Contains an EDIT-P function for using 256 characters.
SYSADDR	SRC	00	009	Findlay, K. Contains entry addresses for MDOS, Mp BASIC, SOLOS, and NORTHSTAR.
S/DISKCOPY	SRC	00	014	Findlay, K. Source for the following substitute for the Mp DISKCOPY. Slower but surer on some systems.
DISKCOPY	SYS	00	004	Findlay, K.

MUG MOD II Library Disk 07, Revision 00

Name	Typ	Rv	Size	Author/Description
====	===	==	===	=====
MENU	BAS	00	005	Rudow, B. This, and the following menus, are unimplemented 'universal' control.
MENU.U	BAS	00	005	Rudow, B.
MENU.A	BAS	00	005	Rudow, B.
MENU.C	BAS	00	005	Rudow, B.
MENU.M	BAS	00	005	Rudow, B.
INSTR.TEXT	TXT	00	013	Shapiro, J. Documentation for the two INSTR. programs that follow.
INSTR.GEN	BAS	00	00E	Shapiro, J. Allows generation/read of data files that contain a program's instructions & documentation. Frees memory & speeds up programs. See news'r #15.
INSTR.READ	BAS	00	00B	Shapiro, J.
TOKEN	BAS	00	007	Powers, W. Hunts down the tokens, or codes, used in Mp BASIC. Lists codes & function See news'r #14.
DISKMENU	BAS	00	007	Burkhardt, E. Reads the directory (DIR) file from video mapped memory. Outputs to screen. See news'r #12.
LABELS	BAS	00	007	Burkhardt, E. Same as above, but prints disk labels.
W-2S.DOC	TXT	00	004	Schoenke, T. Documentation for W-2S.
W-2S	BAS	00	00F	Schoenke, T. Produces W2s & W3s for those whose payroll is manual or whose program doesn't do it.
JUMBLES	BAS	00	008	Riding, G. Solves the JUMBLES puzzle contained in many newspapers. See news'r #16.
TAXDEPR	BAS	00	011	Rothstein, M. Generates depreciation schedules for S/L, 125%DB, 150%DB, DDB or Variable method. Also Investment Credit & Bonus depreciation.
CW.DOC	TXT	00	00B	Fait, D. Documentation for the following 8080 assembly language program.
CW.SRC	SRC	00	02D	Fait, D. Converts keyboard input to a CW sequence, via a look-up table, & sends to output port. Receives & times signal from input port, converts to char. via look-up table, & displays it.
CW	OBJ	00	006	Fait, D.
MUGDOC	TXT	00	019	Findlay, K. Documentation for set of three programs below. See news'r #15, MUG Index.
MUGFM	BAS	00	00A	Findlay, K. Printout for MUG>
MUGINDEX	DAT	00	091	Findlay, K. MUG Index in sequential order.
SORTOUT	DAT	00	091	Findlay, K. MUG Index in program-name/topic order.

PER-REM BAS 00 00C Ivey, W. PERSPECT. documen'n.
 PERSPECT. BAS 00 011 Ivey, W. Aids artist or
 draftsman in producing perspec
 tive drawings.
 TEST1-I DAT 00 00F Ivey, W.
 TEST1-O DAT 00 00F Ivey, W.
 CHECKBOOK BAS 00 00C Little, H. Checkbook balanc'g
 MOTION BAS 00 005 Pickert, A. Illustration of
 screen motion. See news'r #16
 RESCUE.DOC SRC 00 016 Bohn, J. RESCUE documentation
 RESCUE OVL 00 004 Bohn, J. Allows unSCRATCHing
 of files. Handy utility.
 SPELL BAS 00 003 Harden, Jon Neat tool for
 learning. Takes list, displays
 word for variable time. Then
 clears screen. You must now
 type the wod correctly.
 INSERT BAS 00 003 Harden, Jim Binary insertion
 subroutine.
 STOCK-MRKT BAS 00 00D Harden, Jon Tool for personal
 stock decisions.

MICROPOLIS BASED SYSTEMS (Part 1)

There should be a few of you folks out there who are contemplating the purchase fo a new computer. The switch of drive manufacturers by Vector Graphic was certainly a blow to those of us who want to be able to transport our Micropolis operating system software to a new machine. Although VG will still deliver a Micropolis system under special order, this seems like an advantageous time to review our other options.

Each month, for the next few months, I'll give a little information on the systems of which I'm aware. The information is a rewrite of product literature provided by the manufacturers, not my personal experience.

COMPAL 8200

The COMPAL 8200 computer system is a microcomputer based data processing system including CPU, memory, floppy disk drives, video display, I/O interfaces, printer, desk and a variety of system and application software.

It is intended for applications in business data processing, word processing, distributed processing, and software development.

The CPU is a Z80 microprocessor operating at 4 megahertz. It runs on the S-100 bus, has a 1K byte PROM and 56K of RAM. Disk drives are the Micropolis dual floppy, 5 1/4", 315K bytes per diskette.

The Video Display is memory mapped with 24 lines x 80 characters. The display supports the full ASCII set (upper and lower case plus special symbols) on a high resolution 12" CRT. The keyboard is detached, has upper and lower case, and a typewriter paired key layout with a 10-key numeric pad.

Besides the one parallel input port used by the keyboard, the COMPAL 8200 has two 8-bit parallel output ports, two 8-bit parallel input ports and three serial ports configurable in EIA RS-232 format for communications or EIA standard peripherals. Baud rates of 50 to 9,600 and software selectable protocols are supported.

For hardcopy, selection includes a 150-char/sec dot matrix printer, or a 55-char/sec typewriter quality thimble printer. Optional peripheals include an acoustic modem for communications over telephone lines, and a sheet feeder for automatic feed of single sheets of stationery and paper.

System software included in the base price consists of a resident monitor, Micropolis PDS (MDOS, LINEEDIT, DEBUG, ASSM, and the other utilities), Micropolis BASIC, and CP/M.

Optional software, tailored for the COMPAL 8200,

includes a Word Processor, General Ledger, Accounts Receivable, Accounts Payable, Payroll, Professional Billing, and Inventory.

For further information, contact COMPAL at 8500 Wilshire Boulevard, Suite 103, Beverly Hills CA 90211, 213/652-2263.



COMPAL 8200

LIBRARY DISK 6 MODS

Those of you have gotten Library Disk 6, Rev. 4 or lower, will want to make a change or two to the EDIT3 file. Insert the following lines to make the I,L,H, and N commands work in lower case, and the K command work dependably: 2355 MOV A,B 3345 POP D

Robert Manderson had already given me the first change. Ken Findlay figured out both of them. If I remember correctly, those of you with Rev. 1 through 4 have line 2355 put in already. You will, of course, have to reassemble to make a new LINEEDIT5.

This disk, by the way, is the most popular of our history. Seems that lots of you are interested in the entrails of Micropolis operations.

A CALL FOR PROGRAMS

There has been some good response to the system software in disk 6. Obviously, there are some sharp programmers out there. Now, how about all of us working together to generate a couple of those utilities that everyone wants. Bits & pieces will be just fine. I'll put the parts together and everyone will gain.

How about a PIP type of filecopy that will allow multiple files to be specified on the call, or allow all type 'xx' to be copied at once (e.g., all type 10 BASIC files)? How about finding a patch for MDOS or RES that will allow duplicate names, as long as the type is different? How about a catalog program for the directories of all one's disks? It's clear that the talent exists. Let's work to make this Mp S/W be second to none.

Next month, by the way, I'll publish lots of inputs received from members on the CONTROL P, page print, and other enhancements to the system. You know what, I really think we're starting to accomplish something.

Oh yes, and you guys working on the public domain FORTH on Micropolis. Let me know what's happening.

STRING/SUBSTRING OPERATIONS

by Ken Findlay

937 Briar Hill Ave., Toronto Ont. M6B 1M1 Canada

One of the most useful operations of our computers is the manipulation of strings. It can also seem complicated for the beginning BASIC programmer. Listed below are a set of common string operations, an equation for that operation, and examples of their use. Play with them a bit. Change the variables. Then I'm sure you will be able to translate the theory into your own applications. Mp BASIC and BASIC-80 (Microsoft) do string manipulation identically, by the way, so you CP/Mers might also find this useful.

1. Select L bytes from string S\$ starting at position P
R\$=MID\$(S\$,P,L)
If-
S\$="KEN FINDLAY"
L=4:P=5
R\$=MID\$(S\$,P,L):PRINT R\$
Then the computer prints-
FIND
2. Replace substring of string S\$ starting at position P by string X\$
R\$=LEFT\$(S\$,P-1)+X\$+RIGHT\$(S\$,LEN(S\$)-P-LEN(X\$)+1)
If-
S\$="KEN FINDLAY"
P=5:X\$="LOST"
R\$=LEFT\$(S\$,P-1)+X\$+RIGHT\$(S\$,LEN(S\$)-P-LEN(X\$)+1)
PRINT R\$
Then the computer prints-
KEN LOSTLAY
3. Insert string X\$ into string S\$ starting at position P
R\$=LEFT\$(S\$,P-1)+X\$+RIGHT\$(S\$,LEN(S\$)-P+1)
If-
S\$="KEN FINDLAY"
P=5:X\$="LOST & "
R\$=LEFT\$(S\$,P-1)+X\$+RIGHT\$(S\$,LEN(S\$)-P+1):PRINT R\$
Then the computer prints-
KEN LOST & FINDLAY
4. Delete L bytes of string S\$ starting at position P
R\$=LEFT\$(S\$,P-1)+RIGHT\$(S\$,LEN(S\$)-P-L+1)
If-
S\$="KEN FINDLAY"
L=4:P=5
R\$=LEFT\$(S\$,P-1)+RIGHT\$(S\$,LEN(S\$)-P-L+1):PRINT R\$
Then the computer prints-
KEN LAY
5. Add string X\$ to end of string S\$
R\$=S\$+X\$
If-
S\$="KEN FINDLAY"
X\$=" IS FINISHED"
R\$=S\$+X\$:PRINT R\$
Then the computer prints-
KEN FINDLAY IS FINISHED

.....

DRIVE BELT SLIPPING?

by David C. Milne

Cinder Street - Box 6, Oxford, NJ 07863

I/O ERRORS can be caused by drive belt slippage. When it gets bad enough (which can happen almost overnight), the disk will stall when the head loads, and your software may hang. You will know you have the problem when the swish-swish of the disk in its envelope stops upon the click of the head loading. You may have to reset your cpu to get it started again.

Before you panic and send the drive off for repairs, try turning the drive belt inside out. I was able to remove the belt with a hook tool made by bending a piece of solid wire, turn it inside out, and reinstall it in 5 minutes. Add 10 minutes

to get the case apart and back together again. Be careful not to stretch the cloth-reinforced belt. If it still slips, you need a new belt.

I talked with the factory about this, and apparently the problem is known to them. My system was only a month old at the time, so wear isn't the problem.

.....

COMMERCIAL SOFTWARE

THE MICRO LINK

(The following is from vendor specification sheet, not personal experience.)

After you have prepared a letter, invoice data, or virtually any disk file on your microcomputer, would you like to deliver it over the telephone? Would you like to call up The Source or other data bank and review your selection of news afterwards? Is there a file you want to transfer from one computer to another?

Whether you want to send information to the office, link up at home to data banks, or exchange files with owners of other computers (any brand), The Micro Link by Wordcraft claims the ability to do it.

EASY TO USE

The Micro Link is menu driven. It displays the communication on the screen in word-wrapped form. You can set the screen width - 40, 64, 80 characters - whatever your terminal requires.

The Micro Link will bring a disk file - text, data, or program - into memory at any time and send it out. You can save the memory record of the communication to disk, too. Just assign it a file name and The Micro Link puts it away.

A Quick Reference sheet in the middle of the 32-page manual has all the commands.

MEETS YOUR NEEDS

Simple to start, The Micro Link's standard settings are almost all variable under user control. More than 40 commands allow you to host other computers, allocate memory between the copy and filesending buffers, add line feeds or not, and more.

ALL control codes are available for transmission. For computer systems that require line prompt handling, The Micro Link sends a file line by line, pausing for any prompt character and sending the next line upon prompt.

You can interrupt file sending at any place and resume from there. The Micro Link responds automatically to XOFF/XON (control-S/control-Q), too.

This slim program takes up only 5K. That leaves lots of room for call memory. You can turn the call memory on and off quickly, saving only desired portions.

There's a built-in hex and ASCII dump command in case you want to examine memory. The Micro Link sits in high memory, where it can be linked to other programs.

IT'S INSTALLED

The Micro Link arrives ready to go for Micropolis DOS systems. Wordcraft provides you with a half-page fill-in form on your system. Complete it with your dealer or from your equipment manual. Wordcraft delivers The Micro Link set up for you.

HOW TO ORDER

The Micro Link requires Micropolis DOS (or CP/M 1.4, 2.x) and a serial port.

Order from your dealer or send check for \$89 to Wordcraft, c/o Microcomputer Software Associates, 1122 B Street, Hayward, CA 94541, 415/534-2212. Manual only, \$20, credited. Californians add 6% sales tax.

ABOUT MODEMS

Any full duplex, asynchronous modem will work. By special arrangement with Kesa co., Wordcraft offers the DataSpeak O/A-300, a high quality direct connect, full duplex, originate-answer 300 baud modem, for only \$129. It plugs into your telephone without affecting voice use.

.....

LETTERS

Buzz, Perhaps some of the members would like to know about a small Australian outfit called System Software. In particular, they have a program called E.D.O.S., which is a link between Micropolis disk and Exidy Sorcerer ROM PAC BASIC. I have a copy and it works well. They also have a Machine Code Tutorial Package, which is very good. Both, of course, only are appropriate to a Sorcerer, but there may be some MUG members who use a Sorcerer in conjunction with Micropolis disk. System Software also has Screen Genie, CP/M 2.2, Z-SID, a Touch-Type Tutor, a BASIC Tutorial Package, and several game packages - Games Pack 1, Galaxians, Sorcerer Invaders, and Grotnik Wars - all available on Micropolis disk for the Sorcerer, in either MOD I or II format.

Contact System Software at 1 Kent Street, Bicton, Western Australia, 6157. Phone ISD (619) STD (09) 339 3842. They also have dealers in Holland, England, and West Germany, as well as other places in Australia.

I have no commercial attachment with System Software, and have passed the information on just on the possibility that there are MUG members who may benefit from it.

John Donaldson, MYTTON RODD LTD. 770 Lorimer St., Port Melbourne, 3207 Victoria, Australia

.....

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
=====

CLASSIFIED

FOR SALE: Vector Graphic MZ, 64K, Bitstreamer II, Hazeltine 1500, CPM 2.2, all Micropolis software. \$4200.

Joe Castaldo, 213/373-6646
916 Via Nogales, Palos Vedes CA 90274
.....

WANTED: Copy of Flashwriter II, Rev. 2, manual.

Marlin Weston, 303/669-2054 4060 South Garfield Ave., Loveland CO 80537
.....

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
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, C/O Special Systems Technology, Inc
3533 Old Conejo Rd., Suite 111, Newbury Park, CA 913
.....
11/01/81