

MUG Newsletter # 14 - September 1981

\*\*\*\*\*

WRITING GOOD PROGRAMS (PART 2)

by Burks A. Smith of DATASMITH  
Box 8036, Shawnee Mission, KS 66208

Last month we discussed the philosophy of good programming practice. In review, the three requirements of a good program are that it works, it is maintainable, and it gets along with people. Keeping this in mind, a program should be planned from the top down, with major functions broken down into successively more detailed functions, and some sort of written plan is absolutely necessary for anything but the most simple of programs.

Top-down design of a program lends itself to subroutine oriented code. Carried to the extreme (and the extreme may be desirable), the main body of the program consists primarily of GOSUB statements with subroutines performing all the logical functions. Subroutines called by the main body of the program may, in turn, call other subroutines, each of which perform a definable logical function.

If the program has been planned using a flowchart or other planning tool, it should be obvious as to how the code should be grouped in subroutines. Typical subroutines might perform a logical set of calculations like State tax in a payroll, an input operation like getting names and addresses for a mailing list, various output operations, a table lookup routine, etc. The result is a program that is MODULAR in design. What may have been a single long, rambling piece of code is actually a collection of short, easy to understand sub-programs held together by an equally short main program segment consisting of GOSUB statements and some simple logic.

Many programming teachers and textbooks tend to give the impression that subroutines are for saving space when a routine needs to be performed at several different places in a program. While this is true, a far more important function is to make the program easier to understand, more reliable, and maintainable. Even if a subroutine is called only once in a program, its use is justified because it makes the program structure more apparent to the programmer. The space saving for a subroutine that is called repeatedly can be thought of as a bonus.

The use of modular subroutine-oriented code does not automatically make a program a well-written one, and the use of subroutines is invisible to the user of the program. However, if a programmer uses the top-down design approach and modular code, he or she is less likely to introduce bugs in the program due to bad planning and development time should be reduced. Subroutines also improve maintainability, since if a problem should appear or if modifications need to be made, the changes need only be made to a subroutine, which is more or less portable. Productive programmers maintain a library of subroutines they can use in new programs to cut development time and increase reliability.

Second only to the use of subroutines in making a program easy to understand is the liberal use of comments. Even relatively easy to understand code like BASIC quickly becomes just so much jibberish when you have forgotten what it is supposed to be doing, and it may only be a matter of hours until you forget. I have adopted the policy of beginning each subroutine with a comment line (REM or !) that indicates what the subroutine does and adding other comments where needed. The comment line is the entry point of the subroutine so if I see a GOSUB 400 somewhere in the program, I know that line 400 will be a comment line that tells me what that subroutine does.

I decided to try and put BASIC/S through a combination of loops, subroutines, calculations, and string manipulations. The program listed below doesn't accomplish anything - it's just a benchmark test vehicle.

```

10000 !TEST.B/S
10010 SIZES(6,5,49)
10020 PRINT "START"
10030 RS="THIS IS A TEST"
10040 J%=1
10050 FOR I% = 1 TO 1000
10060 ON J% GOSUB 10120,10130,10140
10070 J% = J% + 1
10080 IF J% = 4 THEN J% = 1
10090 NEXT I%
10100 PRINT "END"
10110 END
10120 GOSUB 10150:RETURN
10130 GOSUB 10150:RETURN
10140 GOSUB 10150:RETURN
10150 Q% = (((I%^2) \ 2) * 1.5) + 7) -J%
10160 S$ = MID$(RS,(2 *J%)-1,J%+2)
10170 ON J% GOTO 10180, 10190, 10200
10180 RETURN
10190 RETURN
10200 RETURN

```

It took 57.8 seconds to complete the program under BASIC/S. With Micropolis BASIC, it took 175.4 seconds. BASIC/S ran a little over three times faster. I know that, with Micropolis BASIC, the larger a program is, the slower it runs. So I put the test program on the back of a 300 line program, inserting a GOTO 10010 as the first line. I recompiled the same large program for BASIC/S. When run, it still took 57.8 seconds for BASIC/S. Program size has no effect on execution speed for the compiled version. For Micropolis BASIC, it now took 763.4 seconds. In this benchmark, running identical programs, BASIC/S runs 13.2 times faster than Micropolis BASIC.

The savings of memory space isn't as big a plus as I thought it might be. The BASIC/S version, system and large program, took 37237 decimal bytes. The Micropolis version took 38502 bytes. By compacting Micropolis BASIC, you could get it down to 36710. Compacting the program would have further reduced the Micropolis size.

Last month I mentioned code security as one area of comparison. Interpretive Basics, like Micropolis', have no security. The current operator can change or copy code as he wants. This is a problem for both the software vendor and for the business owner. Vendors hesitate putting in the time to create sophisticated software that someone can easily change and then resell as his own. Employees of business have been known to change, purposely and inadvertently, a line of code. This can result in destroyed data, or worse, erroneous data which isn't discovered for months. While a compiled Basic doesn't provide perfect security, it makes it very difficult for either of the above situations to occur.

My assessment so far - Well, as Don Warner, who also has an "original" version, said to me as we were discussing a few of the bugs, it has taken a bit of the pioneering spirit to get through the first months. Personally, I'm having fun. To be truly easy to convert existing software, one will have to wait for Version 1.1 which will support the DDIM function and a bunch of new ones that don't exist in Micropolis BASIC.

DDIM allows you to dimension an array dynamically, eg, A%(SIZE(1)). There is no distinction between A%(100) and A%(SIZE(1)) in Micropolis BASIC and I use dynamic dimensioning a lot. Bob Zale says that it should be available by the time you receive the newsletter. Owners of Version 1.0 will receive 1.1 for, at most, the cost of materials. Bob also says that Version 1.1 will execute faster than 1.0.

CP/M TECHNICAL TIPS

AUTO-EXECUTE AFTER WARM OR COLD START

by S. Tattersall, ITT  
London Road, Harlow, Essex, England CP179NA

CP/M version 1.4X on Micropolis/Sorcerer includes a feature to execute a program/command on start-up. A 'mode-byte' in CP/M has to be set to enable this feature and is located at 24FFH in SYSGEN. This 'mode-byte' is usually set to D0H (Feature disabled). If it is changed to D1H then a program can be executed on cold start and if it is changed to D2H then the program will execute on warm start.

As distributed, the auto-execute facility will execute a program named AUTO.COM. This program/command name may be altered in SYSGEN at location A08H.

```
0A00 C3 55 03 C3 51 03 7F 04 41 55 54 4F 20 20 20
      20 .U..Q...AUTO
0A10 20 20 20 20 20 20 20 20 43 4F 50 59 52 49 47
      48 COPYRIGH
0A20 54 20 28 43 29 20 31 39 37 38 2C 20 44 49 47
      49 T (C) 1978, DIGI
0A30 54 41 4C 20 52 45 53 45 41 52 43 48 20 20 00
      00 TAL RESEARCH ..
0A40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 .....
0A50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 .....
```

The auto-execute command/program name can be any valid command line, e.g.  
RUN PROGRAM  
or DIR

The program/command name is inserted into the input buffer starting at address A08H, in SYSGEN, and the new program/command name length must be inserted in A07H.

All alternations to CP/M are made in DDT: after loading the configured version, e.g.  
DDT CPM38.COM

and SAVED in the usual way, e.g.  
SAVE 40 CPM38A.COM  
.....

TABLE HANDLING TECHNIQUES

by N. P. Dembinski  
8618 Essex Ave., Chicago IL 60617

Many of us think up programs, begin writing code and before too long, we're bogged down with inefficient program logic that may be difficult to maintain if future revisions are required. If so, it's possible that you are a person in need of programming techniques.

The one most important step in writing a program, often neglected by many programmers, is thinking the logic through in advance.

For your personal project, you're not going to worry too much about program memory size or execution time. But there are a number of techniques that have been developed that will be of benefit to you in writing and maintaining good logic and therefore efficient programs.

Table handling of data is probably one of the most abused techniques in programming. The types of data, frequency of use and size are usually not well thought out in advance.

When reading in data supplied by a user to build an internal table:

1. Check to make sure the data does not exceed the space allocated for the table.
2. Check the sequence if data must be in sequence.
3. If the data contains the subscript determining

its position in the table, check the subscript for a valid range.

Try to find out why you abended (abnormal end of program execution) when a data record came in with a negative subscript that was not checked.

When testing for the end of a table, use a named value giving the item count, rather than using a literal. Then, if the table must be expanded, only one value needed be changed, instead of all references to a literal.

If you are doing a lot of processing on an item in a table, move the item (subscripted) to a work area; do all the processing necessary, and move the item back to the table. Even when subscripts are in computational mode, subscripting takes time.

Table look-up research on the comparative efficiency of four different types of table handling techniques is summarized below.

The tests involved:

1. Reading a file of 21 records to fill a table (in the literal test, records are read, but no table is used).
2. Making 10,500 accesses to a table of 21 entries evenly distributed over the entire table range.
3. Adding the corresponding table data on each access to a control total.
4. Printing a single line (the control table).

TABLE LOOK-UP EXECUTE TIME RATIOS

	DIRECT TECHNIQUE	LITERAL TECHNIQUE	BINARY TECHNIQUE	SERIAL TECHNIQUE
RATIO	1	1	3.7	4.28

NOTE: That as the table size increases from 21 entries tested above, the serial technique would appear relatively less efficient and the binary technique approach would improve efficiency.

The ratios shown do not include compilation, load or set up. They only reflect processing described above in the tests.

1. Use indexing rather than subscripting.
2. Include search argument in sort key if possible.
3. If the table items are subject to seasonal variation, have the program do a count of the transactions each time the program is run, and write it out to tape or disk ready for the next production run.
4. Choice of table organization and access method depends on the programming problem to be solved.
5. Use data analysis - check for strings of equal data and go against the table once.
6. When tables in memory are so large that it must be kept on a disk, build a skeletal table in memory with disk addresses of the full record and find records by direct access;

OR

Build part of the table in memory - those that are most frequently referenced - and keep the rest on disk.

OR

Sort transactions against the table and bring part of the table into memory at a time.

This summarizes some of the programming techniques and guidelines useful for efficient table handling. Next month I'll give a detailed description of each technique.

.....

CDS VERSATILE SERVICE

Tim Dawalt passed me some information about repair facilities for the defunct CDS Versatile computer, which, of course, has the Micropolis S-100 system. I spoke with Don Smathers, the president of

Computerease, and verified that he has plenty of parts for the CDS.

He also has some application software for CP/M on Micropolis. Among the selection are a General Business Package, a Dental Package (\$990), and a Temporary Placement Package (\$2500).

The CDS Versatile was originally sold with a Micropolis BASIC Business Package. Don has no knowledge of the legal status of this software, though he can't recall seeing a copyright anywhere in the code. You guys with this software should take a good look at it. If it really doesn't have a copyright, you might consider sharing it.

At any rate, if any of the members has a problem with their CDS, contact Computerease, P.O. Box 4156, Wilmington Delaware 19807, phone (302) 654-6775.

.....

#### PDS VERSION 4.0

Some of you are still running under Version 2 or 3 of the Micropolis Program Development System (PDS). Version 3 and Version 4 BASIC programs look the same, that is, they'll run on each other's PDS. Version 4 BASIC executes faster, however, and also has editing capabilities not contained in Version 3.

The biggest change is that the RES and MDOS areas are different for the two systems. The MUG Library's assembly language routines won't run on Version 3.

You can obtain Version 4 from Micropolis, Order Processing Dept., 21329 Nordhoff Street, Chatsworth CA 91311 - phone (213) 709-3300. If you just want the disks, the cost is \$25. A new manual costs another \$50, or \$75 for the full package. Since a third of the manual is changed, I'd say you should get it if you get the disks. Postage costs will be added to the above prices.

.....

#### WRITING GOOD PROGRAMS (cont. from pg 1)

Using comments as entry points helps to avoid accidentally removing an entry point when a program is modified by removing lines or changing line numbers.

Comments in a program use up memory and the programmer must attempt to strike a balance between program size and readability. In reality, however, program size is not a very serious consideration in commercial business applications programs, because the computers involved usually have ample memory. This is not always the case with hobbyists with a small budget, but in general, memory can be considered to be cheaper than good software within reasonable limits.

One technique for writing application programs for sale is to develop the code on a computer with a large memory size and fully comment it during the development stage. Then, when the programs are debugged and ready for sale, use a program that removes the comments (and spaces) from the code, such as Systemation's CRUNCH or Datasmith's SMASH. It is not unusual to get a program size reduction of 20 to 40 percent with these programs, and the condensed code can be distributed while the fully commented version is retained for use in maintenance and further debugging.

I wrote the program below a year or so ago to help myself figure out some memory dumps one afternoon. It converts a number entered by the operator into its equivalent notation in any number base desired. When I found it recently by accident, it took me quite a while to figure out what it was supposed to do, probably nearly as long as it took to write it. It works, however, and is representative of a short

program that was intended to be used only once. Note that it is an endless loop and takes a Control-C to stop:

```
10 DIM A$(80)
30 INPUT "ENTER A NUMBER"; X
50 INPUT "ENTER CONVERSION RADIX";B
70 R=MOD(X,B)
80 X=INT(X/B)
90 IF R<10 THEN A$=FMT(R,"9")+A$: GOTO 110
100 A$=CHAR$(ASC("A")+R-10)+A$
110 IF N>0 THEN 70
120 PRINT FMT(B,"Z9R ") +A$
130 PRINT
140 GOTO 30
```

The following is the same program, using nearly identical code, that I modified for this article. I've gone a little overboard for such a short program to illustrate a fully documented, modular program. The only feature that has been added is a way to stop it, and the INPUT statements have been made more descriptive of what they want from the operator:

```
10 ! **** CONVERT ****
20 ! CONVERTS NUMBERS TO ANY BASE.
30 ! WRITTEN BY BURKS A. SMITH 6-20-81
35 ! (REVISION 0)
40 !
50 DIM A$(80): !STRING FOR HOLDING ANSWER
60 PRINT "NUMBERS MAY BE ENTERED IN DECIMAL
  (xxxx) OR RADIX FORMAT (bbRxxxx)"
70 !
100 !< MAIN PROGRAM
110 !
120 GOSUB 200: !GET INPUT FROM OPERATOR
130 IF X=0 THEN STOP: !QUIT IF OPERATOR
  ENTERS ZERO
140 GOSUB 300: !PERFORM CONVERSION
150 GOSUB 400: !PRINT ANSWER
160 GOTO 110: !AND LOOP
170 !
200 !< GET INPUT FROM OPERATOR
210 !
220 INPUT "ENTER NUMBER TO CONVERT (0 TO QUIT)";X
230 IF X=0 THEN RETURN: !RET IF OPERATOR WANTS
  TO QUIT
240 INPUT "ENTER RADIX FOR CONVERSION";B
250 IF B<2 OR B>36 THEN PRINT "INVALID RADIX"
  :GOTO 240
260 RETURN
270 !
300 !< CONVERT NUMBER TO DESIRED BASE
310 !
320 A$="" : !CLEAR THE ANSWER STRING
330 R=MOD(X,B): !REMAINDER OF NUMBER/BASE IS A
  DIGIT OF THE ANSWER
340 IF R<10 THEN A$=FMT(R,"9")+A$: GOTO 360: !USE
  NUMBER IN ANSWER FOR R<10.
350 A$=CHAR$(ASC("A")+R-10)+A$: !USE LETTER FOR
  R>=10.
360 X=INT(X/B): !COMPUTE NUMBER FOR NEXT ITERATION
370 IF X>0 THEN 330: !LOOP IF MORE TO DO
380 RETURN !ELSE RETURN
390 !
400 !< PRINT THE ANSWER
410 !
420 PRINT FMT(B,"Z9R ") +A$: !PRINT THE RADIX AND
  ANSWER
430 PRINT
440 RETURN
```

Obviously, the second version of the program is easier to read for a human and is more or less self-documented due to the large number of comments. The extravagant use of memory for comments is of no consequence here because we aren't about to run out, but some kind of balance has to be struck on "serious" programs. Using a comment as the first line of each subroutine has the highest priority and comments connected with trivial statements has the lowest.

These programs are presented to illustrate a style of coding programs that improves eye-appeal and readability. Hopefully, this will also improve the development of a logical program and the ability of the program to be maintained over a long period of time. NEXT MONTH I'll discuss some human factors,

and what should be considered when writing programs that have to get along with people.  
 .....

ED NOTE: Burks has a nice article on "Using Micropolis Basic" in the September Issue of Interface Age. Pick it up for a little extra education.  
 .....

#### CORRECTIONS

The August newsletter spoke of Vector Graphics with Tandem drives. The drive manufacturer is Tandon, not Tandem. Tandem makes computer systems, which, I believe, use Micropolis drives.

The July newsletter has an article on modifying CBRK to use the space bar instead of Control S for PAUSE. I incorrectly assumed that everyone's CBRK routine was the same. They aren't. I'll find the correct locations for as many systems as I can, and tell you the results next month.  
 .....

#### READING "DIR" FROM BASIC

Although Bob Zale (Systemation) has been up to his ears in BASIC/S, he gave some information that allowed me to dig a bit deeper into processing of the DIR file from BASIC.

DIR is stored on track 0, sectors 3 through 12. Each of these ten sectors contains information on 16 possible files. For single sided drives, only the first 8 files are used. The directory therefore has a maximum size of 80 files (10 times 8), but only 77 (MOD II) or 35 (MOD I) can be used. That's because disk space is allocated to a file one track at a time. For each sector, the 16 bytes for each file are allocated as follows:

```
Bytes 00 thru 09 - ASCII name
10 - Starting track number (plus 128)
11 - File type
12 - ?; unknown use
13 - Number of sectors used (plus 1)
14 - ?; unknown use
15 - ?; unknown use
```

Bob told me that RES has a read buffer at 16R3B8. Using this information, I wrote the following program which reads and displays DIR.

```
10 ! TESTDIR
20 OPEN 1 "DIR"
30 EOF(1)=16
40 FOR L%=3 TO 12
50 GET 1 RECORD L% A$
60 GOSUB 130
70 NEXT L%
80 EOF(1)=1
100 PRINT
110 CLOSE 1
120 END
130 J%=16R03B7
140 FOR K%=1 TO 128 STEP 16
150 IF PEEK(J%+K%)=255 THEN GOTO 240
160 FOR I%=0 TO 9
170 PRINT CHR$(PEEK(J%+(K%+I%)));
180 NEXT I%
190 PRINT TAB(12);
200 FOR I%=10 TO 15
210 PRINT FMT(PDEEK(J%+(K%+I%)),"999");" ";
220 NEXT I%
230 PRINT
240 NEXT K%
250 RETURN
```

Line 30 opens the DIR file size to 15 sectors. EOF is always one more than the file size. I don't know why DIR's size is normally zero; it should be 16. But you can't read it if it's zero (that's probably why it's zero), so I had to open it up. Actually it should read EOF(1)=17, but I wasn't

sure whether that would allocate another track, which I didn't want to do. When I'm done (line 80), I set the size back to 0.

Line 50 looks like it puts something into A\$. It doesn't. The data size is greater than 250 bytes, so BASIC ignores the input. The GET places the data in the buffer at 3B8H, though. Line 140 breaks the data up into eight 16-byte pieces. The last half of the data is blank, as I mentioned above. Line 150 checks the first byte for a FFH. This value means that the location is unused. I jump out if that's the case. You ought to put a "!" on this line and run it once on a disk where you've scratched some files. Interesting. Lines 160 & 170 print the ASCII file name. Lines 200-210 print the numeric values in the next six bytes.

A listing by FILES would give a print that looks like this:

```
DIR      03 0000
RES      03 0014
MDOS     0C 001A
```

The values are printed in hex. TESTDIR prints the values in decimal, and its output looks like this:

```
DIR      128 003 000 016 001 000
RES      129 003 000 021 001 001
MDOS     131 012 000 027 001 001
```

CAUTION!!! Whenever you are going to play with data on a disk, practice on a backup. While I don't think this program destroys anything, be careful. I have used it, even left the size of DIR at 16, and everything still works fine.

There are lots of things one can do to enhance the program. Those of you who work on this, let me know what you come up with. If you know the use and organization of the sectors preceding and following 3-12, please tell me about it.  
 .....

#### LIBRARY DISK 6 - SYSTEM SOURCE

Disk 6 of the MUG Library is ready for release. It is the disassembled and commented code for RES and MDOS, a modified version of LINEEDIT, and a Z80 disassembler. Robert Manderson (C/O Hewlett Packard Pty Ltd, 31-41 Joseph St., Blackburn, Victoria, Australia, 3130) did the work. RES and MDOS may not be identical to your system as it seems that all manufacturers have their little changes. The RESI/O module is, of course, normally different for all of us. It would have to be edited for your system. It assembles back to Robert's system, though. In addition to the source code, there are text files of explanation. This is a wealth of information. Robert has done a tremendous job.

The LINEEDIT program is based on Micropolis', but has a MERGE command, an altered APPEND function, and enhanced EDIT capabilities. The Z80 disassembler is a Manderson creation.

The distribution of this disk must be controlled. We must take care not to violate the copyright. It therefore will not be "sold". It will be distributed only on a basis of you submitting a program, disk, and the standard \$3 (North America) or \$5 (other) cost for postage & handling. I must also insist that your submission disk include a Version 4 RES & MDOS in addition to the program submitted for the library. This will prove that you are an owner of Version 4, and that I am not interfering with Micropolis' sales of this product.

Those with MOD I will have to send 2 physical disks. This is a lot of data. Robert has even more, so we'll be expanding on the data base.  
 .....

BASIC TOKENS

by William D. Powers  
3520 San Semeon Ave., Oxnard CA 93033

Those of you who have looked at the code in memory when a BASIC program is loaded, have noticed that the 'key' words are never there. That's because Micropolis BASIC converts these words (PRINT, GOTO, etc.) into one byte tokens. The following program finds and decodes these tokens.

The assumption is made that a token is a byte with a value X, where X is greater than 80H and less than FAH. Each token represents a string of 1 to 9 characters. The program also assumes that there is a table in BASIC of the following form:

```
bbbbXbbXbbbbbbXbbXbbX ...
```

where 'b's form a string of BASIC ASCII characters and the X is the token representing the preceding string.

That such a table exists was shown by examining memory with the MDOS DUMP command, after BASIC had been loaded and control then passed back to MDOS. This examination also showed that the first ASCII string was 'CONT'.

The BASIC module is searched from its start at 1598H until the table is found. Since, for most systems, it starts around 3000H, the search may take a bit of time. The first and last locations of the table are saved, and a memory file is constructed of the data between these locations.

Variable definitions used in the program are:

```
A( ) TOKEN VALUE
B$( ) STRING REPRESENTED BY TOKEN
W TEMPORARY VALUE
N LOCATION START OF TABLE
L EVENTUALLY LAST LOCATION OF TABLE,
MOSTLY, LOCATION TO BE PEEKED AT
```

```
010 !TOKEN
020 DIM A(128),B$(128,10)
030 I=1
040 L=16R1598-1
050 L=L+1
060 IF PEEK(L)<>ASC("C") THEN 50
070 IF PEEK(L+1)<>ASC("O") THEN 50
080 IF PEEK(L+2)<>ASC("N") THEN 50
090 IF PEEK(L+3)<>ASC("T") THEN 50
100 A(I)=PEEK(L+4):B$(I)="CONT":N=L:L=L+4
110 PRINT "TOKEN FOR ";B$(I);TAB(22);"IS ";A(I)
120 I=I+1:B$(I)=" "
130 L=L+1:W=PEEK(L)
140 IF W<128 THEN B$(I)=B$(I)+CHAR$(W):GOTO 130
150 A(I)=W:IF W<>16RFA THEN 110
160 PRINT "TOKEN LIST STARTS AT ";N;" DECIMAL"
170 PRINT "TOKEN LIST STOPS AT ";L;" DECIMAL"
180 END
```

Note that the literal at line 70 is the letter "oh", not a zero. If you want to see what the token looks like, append  
;CHAR\$(A(I))  
to the end of line 110. The interpretation will differ, depending on the type of character generator you have in your system, and will, in most cases, completely disrupt the sequential listing you would normally get.  
.....

LETTERSCP/M USER'S GROUP

Buzz,  
I thought it might interest you to know that rumor has it that SIG/M, which is a CP/M special interest group in this area, is planning to have the CP/M User's Group library on 5" disk - some on Micropolis and some on North Star.

You might want to talk to Hank Kee; 42-24 Colden St., Flushing, NY 11355 (212) 539-3202 to see if MUG can co-operate.

BASIC FUNCTIONS

Also I've run across an interesting use of the SGN function in BASIC. The statement was ON SGN(X)+2 GOTO 100,200,300.

Another thing that I've discovered is that you can do something like FOR I=1 TO SIZE(n) where SIZE(n) is the number of records in file(n), eg, OPEN n "name".

Has anybody got a routine to patch into MDOS to put out an address header prior to a dump from memory?

HARDWARE STANDARDIZATION

Also does anyone know if there is any standardization among drive manufacturers in the pin connections of the cable header to the drive controller board? Can Micropolis drives be made to work with a North Star controller (single density)?

I, too, have had uniformly good results with all Systemation products in addition to having Bob Zale be generous with his time in resolving numerous difficulties with my application programs.

Keep up the good work!

Martin C. Rothstein, CPA  
21 E 40th St, New York NY 10016

Marty: I'll check out SIG/M and a couple other CP/M groups and write the results next month.

Are you aware of the fact that you can use a literal on most MDOS commands? Specifically, you can do an ASSIGN 2 3 to get your output to the printer and -

```
>DUMP 04E7 04F0 "09/01/81"
>FILES "09/01/81"
```

The literal must be 10 characters or less.  
.....

UPGRADE FROM MOD I TO MOD II

Buzz,  
I read an article that said the difference between single density and D.D. was that D.D. omits clock pulses and uses more compact coding format. As a result, the data transfer rate doesn't change. I don't think that is true of Micropolis MOD.1 / MOD.2. My question then is - ...I have MOD.1. What is cheapest way to upgrade to MOD.2? The answer that I expect is - sell MOD.1 and buy MOD.2.

Murray MacKenzie  
38 Inniswood Drive, Scarborough, Ont Canada

Murray: From a data transfer viewpoint, there isn't any difference between MOD I and II. Both drives turn at 300 RPM, have 16 sectors/track, of 268 (unformatted) bytes/sector, and transfer data at a 250 Kbit/second rate. The MOD II has 77 tracks, where the MOD I has 35.

You don't have to sell your MOD I to go to MOD II. The controller board (the S-100 board in your computer) is the same for either type drive. The software is the same (however, see the Jack Rice letter, below). MOD I and II drives can be mixed in any sequence among the four drives serviced by a single controller board.

I have MOD II's as drives 0 and 1, and MOD I's as drives 2 and 3. You would simply need the add-on modules and a new 3-(or 4) connector cable.

The cost of a single-sided, single MOD II add-on drive (MCP-1023M2) costs \$495 from Priority-One. A double drive costs \$895. You'd also need the Daisy Chain cable (MCP-1083-02) at \$40.

You can get the full new system, which includes controller, manuals and system disks, for \$695 and \$995, respectively. You'd still need the daisy

chain if you wanted to use your MOD-I. Call Priority-One, toll-free, at (800) 423-5922.  
 .....

#### DISKS THAT WORK

Buzz,  
 I would find a list of disk types (manufacturers) and part numbers that work with MOD II most useful. I can no longer buy disks here that work with my Micropolis, and when ordering by mail, frequently going through customs, it would be useful to be sure of getting a diskette that would work with 77 tracks, the first time.

Doug Ellis  
 PO Box 485, Halifax NS

I know that there are lots of disks which work. Keeping the product numbers straight for each brand is a chore, though. Lately I've been using the new Verbatim DATALIFE disks, with good results. I've had no trouble with the 'MD 550-16-18209 double sided/double density 16 sectored, certified 40 tracks per side' disk. I haven't tried the MD 525-16, a single sided version of the above, because that's the same part number of their old single density disk that I never liked.

To be really sure, get one of the following:  
 MD 577-16-18230 single sided/double density 16 sectored, certified 77 tracks  
 MD 557-16-18257 double sided/double density 16 sectored, certified 77 tracks/side

If you have any trouble finding these, or their equivalent, I can send you a softbox package of 10 MD 577-16-18230 disks for \$42 postpaid to North America. Again, as for everthing in the MUG, you can use your VISA or MASTERCHARGE but the charge is 104% of the cash price.  
 .....

#### CP/M VS MDOS

Buzz,  
 Keep swinging with the MUG.

We run a lot of CP/M at this firm, as you might know, and we are not running so much MDOS at all.

Anyhow, thanks to your MUG magazine we learned about the CP/M-MDOS translation thing from Systemation Inc. I think that Systemation Inc. should tell Lifeboat that there is such a utility and it should be put up on Lifeboat's price-list.

We probably have more interesting U.S. software for CP/M than anybody else in Sweden.

This CP/M-MDOS is a must for our firm.

We develop software in Assembler very often under MDOS ASM because we find it better to work with than e.g., CP/M MAC that we also have. Thus we develop software with ASM LINK1 and LINK2 and debug it with DEBUG-GEN.

When it works we transform it to 84-files with TRANSLATOR II and then put it on CP/M with UTL-1 package from Systemation Inc..

The best thing with MDOS is that you can run a short piece of assembler under BASIC for debugging and test running.

Having thus CP/M and MDOS you have the best of all worlds.

I myself prefer CP/M operative system and under CP/M we have bought many application and utility programs: Wordstar, the best wordprocessor there is, for a translator of foreign language, PL/I-80, SID, MAC, Datastar, Mail-Merge, T/Maker, a fantastic Tarbell editor - wordmaster, Magsam IV, QSORT, CBASIC, FORTRAN-80, FORTH!!!, SELECTOR-IV for our administration files, GLECTOR for bookkeeping, CAT for catalogizing, BSTAM and DISTEL. It would be possible to write an article

about it, and I did in the Swedish Mikrodatorn. Although MDOS is far better than CP/M in many aspects, MDOS could not offer all this software that we have purchased and that we really need. Moreover MDOS op. system is not suitable for bulk erasing files and many facilities we use under CP/M as translators. Moreover CP/M software is compatible even if you change to a hard disk.

MDOS is much safer to run and should there be a Wordstar under MDOS I might even consider to work under MDOS with my translations.

I think the hobbyist and the hacker should stick to MDOS, but a business computerist simply has to face facts and run under CP/M environment.

Maybe you don't share our ideas.

In spite of what I have said about CP/M, we remain MUG faithful and send a cheque of the due amount for 1982 and really thank you for the newsletter which is valuable information for us.

Bill Leksen  
 POLKO, Box 12184, 102 25 Stockholm Sweden

Bill: I see your point, and I don't exactly disagree with you, but I have a slightly different point of view. If one is going to purchase a wordprocessor or a Fortran compiler, you need CP/M - at least at this point in time. There certainly is more software available for CP/M.

It isn't as clean as it looks, however. There are multiple versions of CP/M. There are also multiple versions of BASIC, each of those having several revisions. Any application program may require some set of a particular CP/M and a particular revision of a particular BASIC. Other applications programs might require Fortran or one of the multiple versions of Pascal.

With Micropolis, things are pretty static, and therefore considerably less confusing and less expensive. MpBASIC Rev. 3 works on Rev. 4. Revision 4 has been around for years and probably will never get changed. If you, meaning, in general, any member, are going to write your own software, I think you should stay with Micropolis. There is also a fair amount of software that can be purchased. It is all transferrable to the Micropolis hard (Rigid) disk, too.

There are a substantial number of members in the group who are in business, who do own CP/M and a wordprocessor, but who use Micropolis for everything else.

It's not that I disagree with your personal choice of how to run, Bill. I'm sure that it's proper for your circumstances. But I do disagree with the blanket statement that any "professional" system must be primarily, or totally, CP/M. However, I appreciate your comments, and am pleased that you are continuing your participation in the MUG.  
 .....

#### SPECIALTY PROGRAMS

Buzz,  
 I am not really a computer nut, not that I wouldn't like to be, but I just don't have the time. All the software I have is licensed software; I am in a specialty business for which off the shelf software is not available. We are in the Pension/Profit Sharing administration business.

I have developed a few programs on my own (in M.BASIC). One of them is a program that calculates single and joint and survivor annuities at any interest rate and two annuity mortality tables, with cost of living increases at any interest rate after retirement. A combination of any male or female ages can be used. It is strictly an actuarial program and probably not of any interest to anyone not in my type of business.

Another program that I developed is an interest program which produces four columns for any rate of interest and duration as follows:

Present value of \$1.00, present value of \$1.00 a year, accumulated value of \$1.00 and accumulated value of \$1.00 a year. Values can be determined at either the beginning or the end of the payment period. If there is any interest in these I'll be happy to exchange them for something else.

By the way, I have two Vector Graphic Computers, Model II (MDOS and CP/M 2.2), and two Diablo 1620 printers. If I can get one cheap enough I might consider buying an additional one.

Rudolf L. Duering, Duering & Associates, Inc  
582 Market St, San Francisco CA 94104 415-421-5053

Rudolf: I'd love to trade each of your programs for a disk full of software. While each program qualifies you for a full library disk, be sure to include an extra disk of yours for each library disk. I need to use your disks for the \$3 rate.

Specialty programs are what really make systems useful. I wish we could get loads of them. Maybe we should start a set of disks for Accountants, Lawyers, Construction, etc.

.....

#### RUNNING MOD I & II TOGETHER

Buzz,  
I have obtained a new interface cable from Dave Land in Kansas City, and now have my MOD I single drive attached with my MOD II drives (1053) and everything seems to work ok. That is with the expected results that MDOS does not know that the third drive out there is not a MOD II drive like the first two. Type in "FREE 2" and he gives me about 42 more files than I should have. But we can live with that since we now have a way to get all files, both program and data, back and forth between the II's and the single I. I just have to keep the smaller MOD I size in mind when transferring files.

Jack C. Rice  
11411 Hubbard, Kansas City KS 66109

Jack: You are pretty much where I was a couple of months ago. Many people wrote me and told of the configuration bytes for the disks drives which are 0884H, (drive 0) through 0887H (drive 3).

You can see the contents of these by typing (from MDOS) -

```
>DUMP 0884 0887 <cr>
The computer will respond with
0884 4D 4D 4D 4D
For your system, MOD II's in 0 and 1, MOD I in 2,
Type
>ENTER 0886 <cr>
>23/ <cr>
If you had MOD I's in both drives 2 and 3 the
second line response would have been 23 23/ <cr>.
You can save your new RES by having your system
disk in drive 0 and typing:
>TYPE "RES" 0
>SCRATCH "RES"
>SAVE "RES" 2B1 15983
```

I ran into another problem with DISKCOPY. I was trying to copy from drive 2 to 3, both MOD I's. I'd always get errors. The problem was that DISKCOPY was called in off drive 0. DISKCOPY gets its max-track information from the drive it was read from. It was trying to copy 77 tracks instead of 35. The solution was to either call DISKCOPY from drive 2 or 3, or to temporarily set location 084H to 23H.

.....

#### EXIDY'S MDOS CONFIGURATION

Buzz,  
My Sorcerer requires that I leave a 'hole' in S-100 memory for the controller. That's a bit of a problem in that neither board will allow a block smaller than 4K to be moved nor will either board allow any block to be disabled. Since all the controller needs is 1K, it would be nice to disable just that amount. Both are static RAM; one is a Vector Graphics 16K Rev. 2. The other is an Ithaca Audio 8K 1A-1110 Rev. A. Since the Sorcerer doesn't support the Phantom line, I guess I'll have to get hold of CS\* on the block in question. The controller will be running one of three places - normally D400, sometimes BC00, more often B400.

The above question can be summarized as follows: Given that I've no control of PHANTOM\* (pin 67), how do I disable a 1K block for the disk controller?

Don Myklebust  
19710 Guthrie, Strathmore CA 93267 (209) 568-1389

Don: I don't think you need to leave the 'hole'. Contact Bob Hageman at the Sorcerer's Apprentice, 313/535-9186. I believe you can insert the controller in space currently designated for, but unused, by PROM boards.

.....

#### DIRECTIONS OF MUG

Buzz,  
Something I'd like to see is an overview, a look at the whole area of microcomputing and where we are in relationship to what's going on, our percentage of the market in hardware, software, what publications (magazines) are aimed where and what will be important to learn or not. There seems to be a glut of information and a need to have a clear idea of priorities. I know this is beyond the scope of MUG, but it must be a concern for the group.

Listed below is a routine I use to insert a new entry into a alphabetical list or file. It successively approximates the position so that a maximum of log (2) N comparison, are required (10 for a list of 1024, for example).

```
100 O$=N$(N): ! Set dummy holding variable,
105 ! O$, equal to new entry, N$(N)
110 H=N:G=0: ! Set Max position, H, equal to
115 ! file length, N
116 ! Set Min position, G, equal to 0
120 M=INT((H-G)/2): ! Finding increment 1/2 dis-
125 ! tance between max & min,
126 ! (H & G)
130 IF M<1 THEN 160: ! Position of new entry found
140 IF O$>N$(M+G) G=M+G:GOTO 120: ! New entry >
145 ! min+increment, new min created
150 H=M:GOTO 120: ! New entry < min+increment,
155 ! new max created
160 FOR J=N TO H+1 STEP -1:N$(J)=N$(J-1):NEXT J
165 ! Make room for new entry
170 N$(H)=O$:RETURN: ! Place new entry into
175 ! proper position
```

Jim Harden, Harden Farms  
McKnight Rd., Cleveland MS 38732

Jim,  
Gee, a philosopher in our midst. You said a great deal in one paragraph. However, the MUG is intended, or at least capable, of aiding you in all areas mentioned, with the exception of priorities. What's important to me isn't necessarily important to you. I try to cover all the bases. This probably makes parts of the newsletter useless to those who aren't interested in...(name your poison - graphics, communications, assembly language, etc).

There aren't any magazines aimed at Micropolis. The closest are Kilobaud, (S-100) Micro Systems, and Interface Age. Other magazines aim at areas which may be important to you. MUG's specific role is not to compete with them, but to alert you to

their existance. MUG's primary purposes are the documentation of the Micropolis Operating System and the software which works on those systems. We have a rough time doing that!

I believe that Graphics, Communications and Distributed Data Processing are the "growth" areas. The "typed" and "structured" languages, such as Pascal and Ada, are getting a lot of press. Ada is quite complicated, though, and I don't know if a compiler for micros will be created. Don't know if people will want to write in it, either.

There's no doubt that if you want to be a part of what's going on in microcomputers, you better get CP/M. We are definitely a minority. However, it doesn't bother me to not be in the "pack". I never particularly liked Disco or Punk Rock, and don't care for TV game shows. There are lots of instances where I prefer to separate myself from what the majority is doing. Micropolis happens to be one of those instances.

.....

CBASIC LIBRARY?

Buzz,  
I purchased the Osborne general ledger and payroll programs in CBASIC on the MOD II format. As these are considered public domain, would they be suitable for the library?

Leonard S. Darsey, O. F. ECKLUND, INC.  
PO Box 279, Cape Coral, FL 33910

.....

Leonard: We presently have nothing but Micropolis software in our library. I have no problem with establishing a CBASIC (or MBASIC) set of library disks. I wonder about the practicality. I'm under the impression that CBASIC programs won't run on CBASIC II, and in fact, CBASIC II Version 2.06 programs won't run on CBASIC II Version 2.07. Never-the-less, if the membership feels that such

CP/M software is useful, we should certainly start to accumulate the same sort of routines and programs for CP/M languages as we now are doing for the Micropolis languages.

The final point - I don't know how software gets into the public domain. The Osborne software is still being sold, isn't it? The programs would be nice to have but we must be careful and have some certification that they are not under copyright.

.....

CLASSIFIED

FOR SALE: SOL computer system with 48K static memory, dual Micropolis MOD II drives, Panasonic Monitor/TV. Works perfectly and includes lots of software - MDOS + BASIC 4.0, CP/M (Lifeboat + custom SOL), CBASIC2, Electric Pencil I and II, Systemation Utilities, Games, etc., etc., etc. \$2500 for everything.  
Dean Sabins, 9811 Hallard Dr #112, Laurel, MD 20708 (301) 792-0790

.....

FOR SALE: System integrated into a Trimm Industries 3' X 5' single-bay pedestal desk with lap drawer. Contains a TEI rack-mounted S-100, 22-slot mainframe chassis with heavy duty power supplies; Cromemco 4Mhz Z80 CPU, IMSAI SIO2 serial I/O, 56K of contiguous 4Mhz static memory, Intertube CRT intelligent video terminal, Micropolis 1053 MOD II dual drive. Software includes Version 4.0 MDOS, BASIC, etc., Systemation's UTL-1, DSM-1 & BEM, and the CCA Data Base Management system. Complete H/W & S/W documentation. System available by mid-December. \$3000, shipping included.

John Martin, C/O The Redington Group, 6605 S. 135th St., Omaha NE 68137. (402) 895-3296

.....  
09/01/81

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  
=====