

MUG Newsletter # 20 - March 1982

FORTH: A STEAL - FOR A SHORT TIME

by Buzz Rudow

INTERNAL REPRESENTATION OF DATA

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

The mathematical accuracy of Micropolis Basic may very well be the best of any high level language available for microcomputers today. I have not seen any that even come close, and the most popular Basic versions are certainly much worse. By use of the SIZES statement, the user is able to directly specify the number of bytes used to store variables, and the result is floating-point accuracy that can vary from just a couple to as many as 58 significant digits. This is in sharp contrast to other Basics that generally offer only single precision (6 digits) or double precision (16 digits) as the only choices for precision available.

The accuracy of the computer language is not simply a function of the number of digits it can store. It is also dependent on how the data is represented internally. There are two basic schemes for doing this. First, is binary, where numbers are stored as pure binary (base two) format. Second, is Binary Coded Decimal or BCD, where numbers are stored as packed binary representations of decimal numbers. This distinction is important, so we will take each method separately.

As we have discussed previously in this column, a binary number is composed only of zeros and ones, with each digit position representing a power of two. Thus a binary 1 equals a decimal 1. A binary 10 equals a decimal 2. A binary 100 equals a decimal 4, etc. The largest number that can be stored in 8 bits (digits) is 255 and the largest number that can be stored in 16 bits is 65535. By simply providing enough bits, any accuracy that is desired can be obtained. However, as a practical matter 32 bits for single precision and 64 bits for double precision is typical. Pure binary format has the advantage that it is the most compact way of storing a number in a computer and mathematical operations are extremely fast, since a microprocessor is a binary device.

The disadvantage of using pure binary storage and pure binary arithmetic is that humans don't use the binary number system. In almost all cases, input and output is expected to be in decimal so the number must be converted from one base to another on the way. This can introduce errors, which means that the number finally stored in memory is not the same as the number the program read or the number printed is not the same as the one retrieved from memory. The result of mathematical computations may not come out exactly as you expect either, because binary truncation and rounding errors will be different than you expect in decimal.

(continued on page 3)

LATE NEWSBASIC/Z RELEASED

The BASIC/Z compiler was received in mid-February. It is available from the MUG for \$308. I didn't have time to review it before press-time, but will definitely do so for the April newsletter. It looks great. See the article on page 6.

You should also note that Systemation has changed its name, address and phone. Bob Zale's new company is System/Z, P.O. Box 11, Richton Park IL 60471 - Phone (312) 481-8085.

Sometime in the next three months, Acropolis is going to update their FORTH to multi-tasking, multi-user. A best kept secret, until the MUG reveals it now, is that when released, Dr. Shaw, of Acropolis, will give current owners a copy of the new system. He doesn't want to support two systems. Now that's quite a giveaway.

BUY NOW

Here's the deal. Send (or call) a check (or VISA/MC) for \$150 to the MUG now. You'll get one single-tasking FORTH system. When the new multi-tasking, multi-user system comes out, you'll get it FREE!. The new system will sell for \$250-300.

MDOS BASED

Acropolis FORTH runs under MDOS. That means it runs faster and cleaner than a FORTH system under CP/M. CP/M design requires user interface conventions which restrict the ability of FORTH to get directly to the hardware.

DETAILS OF ACROPOLIS FORTH

The FORTH language features the built-in advantages of high speed execution, minimal memory requirements and ability to store more data in less space than other languages. Now ACROPOLIS has written a special version of FORTH specifically for Micropolis users.

A-FORTH operates as its own compiler allowing you to compile directly from the keyboard as well as source files on your disks. A-FORTH compiler security traps errors by insuring all control structures are completed within your program.

A-FORTH is an interactive language that allows creating and testing programs right from the keyboard without creating a disk file. This user-oriented keyboard testing is interpretive in nature to aid in debugging your program.

A-FORTH is a structured language so your programs are more readable. It is completely extensible so you can add your own commands to the language.

A-FORTH is up to 50+ times faster than Micropolis Basic. Even faster execution (up to 600+ times) is possible for time critical applications by using A-FORTH's multi-level language feature to program in assembler. A-FORTH also features an 8080/8085 macro-assembler that allows you to use any mixture of A-FORTH and assembly code you want in a single program. You can even use A-FORTH's macro-assembler to create a whole new set of instructions for your machine. A-FORTH's macro-assembler contains control structures for forward and backward references and is provided in source form so you can make your own modifications.

ACROPOLIS sponsors an A-FORTH contributors software applications library and provides A-FORTH updates & patches at no charge for one full year. Special features are:

- Supports system printer using standard MDOS assigned statements
- Enhanced disk procedures to reduce response & compiling time & # of disk accesses
- Physical disk support for disk diagnostics and disk copy
- Allows access to MDOS file directory
- Can be set up for multiple tasks and users

MUG BACKING

The MUG intends to back Acropolis' implementations of FORTH as a standard for the group. This newsletter will print articles on it. I personally intend to become an "experienced novice".

(Representation of Data - continued from column 1)

Numbers in Micropolis Basic are not stored in binary, but in Binary Coded Decimal (BCD). Of course, this is still a form of binary representation because computers can't do otherwise, but the number remains in decimal format internally. To do this, each digit of the decimal number is stored in 4 bits, with an 8-bit byte holding exactly two decimal digits. Thus 8 bits can hold only up to 99 and 16 bits can hold up to only 9999. While not as efficient as pure binary, there is no translation error and truncation and rounding are handled exactly as you expect them to be. Math operations also remain in decimal, since the 8080 and Z-80 microprocessors support BCD arithmetic through use of the DAA (Decimal Adjust Accumulator), which allows the processors to maintain data in BCD at all times.

Micropolis Basic has three variable types: Real, Integer, and String. The precision of each can be set using the SIZES statement, which is explained in the manual. String variables consist of a series of bytes representing ASCII codes for various characters. Each character in the string uses one byte of memory. The actual string is preceded by two additional bytes. The first tells how long the string can be as a maximum, and the second tells how long the string currently is. It is the value of this second byte that is returned by the LEN function.

Integer variables are variables that do not contain a decimal point. They all use the same number of bytes as specified by the SIZES statement or a default of 3 bytes each, and are in BCD format, right justified. Since the default sizes is 3 for integers, they can normally be 6 digits long.

Real variables should be more accurately called floating-point variables. They are stored like scientific notation with a number part and an exponent of ten to specify where the decimal point is to be placed. Internal format begins with an 8-bit exponent byte with the 8th bit specifying the sign of the number part. The 7th bit specifies the sign of the exponent (set means positive), and the last six bits representing a power of ten. The rest of the bytes specify the number part of the value in BCD with a decimal point assumed to immediately precede the number. This is exactly how the computer will print a value using the E format.

Next month I'll disclose the locations of the pointers to all variables used by a Basic program.
.....

WRITING AND READING DISK FILES

by Buzz Rudow

In computer nomenclature, a file is a set of data which is accumulated under a single name. In a file, one has some set of records. The set can be 0, also referred to as an empty, or null file. Each record has some set of fields. Records within a file do not have to be structured with identical fields, but for this discussion, they will. Looking at the universal example of a mail list, one might have fields as simple as; name, address, city, state and zip.

The same data might require field assignments as diverse as; salutation, first name, middle initial, last name, title, company name, address line 1, address line 2, city, state, zip, country and flag.

However, for the sake of simplicity, I'll direct my discussion to the short set of fields.

Each field is normally represented by a separate variable. In the first example, you might use N\$, A\$, and C\$. Each set of N\$, A\$, and C\$ is a record. The set of records you can reach under one file name is, of course, the file.

To get original data into the record, I would ask the operator to key (type) it in. The program might be:

```
PRINT "Enter Name"
INPUT N$
PRINT "Enter Address"
INPUT A$
PRINT "Enter City St Zip"
INPUT C$
```

Suppose the list you are reading from looks like this.

```
Ronald Reagan, President
The White House
Washington, D.C. 20500
```

If you answer 'Ronald Reagan, President' to the first prompt, you'll get an "excess input ignored" error message, and N\$ will contain only 'Ronald Reagan'. The same problem would occur in the third prompt, where you'd get only 'Washington' in C\$. Both problems occur because the comma is the default STRING DELIMITER.

STRING DELIMITER

The string delimiter is useful, and necessary, for separating several strings out of a long set of characters. You could program your input as;

```
PRINT "Enter Name, address, city-st-zip"
INPUT N$,A$,C$
? Ronald Reagan - President,The White House,
Washington DC 20500
```

BASIC then knows to put everything up to the first comma (string delimiter) into N\$, then all up to the second comma into A\$, and the rest into C\$.

Some of you are now saying to yourself, that's dumb. Why not just input as in the first example and do away with string delimiters and make life simple. Perhaps you have a point when discussing keyboard input, but string delimiters are also used on the disk.

As mentioned in previous issues, Micropolis disks have 77 tracks (35 on MOD I), each track having 16 sectors, each sector having 250 bytes (for BASIC data files). A byte is an eight bit word and is used to store a character.

The 'primitive', or lowest level, disk operator of Micropolis is to read or write a sector. Any time you do a PUT, you'll write 250 bytes. If you wrote the example data to disk as;

```
PUT 1 N$
PUT 1 A$
PUT 1 C$
```

you'd use three sectors. The first sector would contain 'Ronald Reagan - President' followed by blanks. You waste a lot of space.

PACKING DATA

So you try and pack the data for one record into one sector, using something like;

```
PUT 1 N$,A$,C$
```

The comma in the above PUT statement is NOT a string delimiter - it's a variable separator. I know - it gets confusing. The comma does not get written to the disk. In using the comma as a variable separator, you have also told the Disk Operating System (DOS) to start writing the next field at a location that is a multiple of 16.

```
(column #)      2      3
                5      2
Ronald Reagan - President      The
```

An alternative to using the comma as a variable separator is to use the semicolon. The semicolon tells the DOS to start writing the next field immediately after the preceding field. You could

say -

```
PUT 1 N$;A$;C$
```

Then the disk sector looks like

```
Ronald Reagan - PresidentThe
```

At first glance, you might say that we'd have to use the comma. The human eye can see where the lines separate. You can, in this example, but you can't see, as a rule, where the strings actually separate in either method. The name, N\$, could have 31 characters and run right up to location 32, which would be the start of the address. The DOS has nothing to tell it where one stops and another begins.

Now don't misunderstand me. Either of the above PUT statements are legal. There are times when you'll use them, but you'd better have a way to get your data back off the disk.

NOW - TO GET IT BACK

Normally we get data off the disk by saying;

```
GET 1 N$,A$,C$
```

Basic reads all 250 bytes into a buffer, then parses it to the variable you named. Parsing is the function of decoding a set of characters into separate fields. In the default mode, that is, if you just run BASIC without altering the string delimiter, BASIC expects to see a comma, on the disk, at the end of any string. When it tries to parse the data written by either of our above PUT examples, it never sees a comma at all. BASIC therefore puts all the data into N\$ and gives you a "RAN OUT OF DATA" error message.

The proper way to write the data, therefore, is;

```
PUT 1 N$;"",A$;"",C$;"",
```

You physically write the comma on the disk. The disk sector now has;

```
Ronald Reagan - President,The.....
```

When read in by the;

```
GET 1 N$,A$,C$,
```

the parsing sees commas and is able to put the data into the proper variable.

Now we know how it works, but we still have the problem of not being able to use a comma in our data, since the comma has the specific job of being the string delimiter. This can be a small or large irritation, depending on the type of data with which you are dealing.

CHANGING THE STRING DELIMITER

The solution is to change the string delimiter. Most Basics don't allow you to change delimiter, but our's does. Just be sure to pick one that isn't contained in your data. Let's use the exclamation point (!).

The program can now read;

```
STRING "!"
PRINT "Enter Name"
INPUT N$
etc.
```

and you can answer

```
Ronald Reagan, President
The White House
Washington, DC 20500
```

To write the data, do a

```
PUT 1 N$;"!";A$;"!";C$;"!"
```

Sometimes, for convenience, I'll put the string

delimiter into a string variable and write the following code which is operationally identical to the statement above.

```
Z$="!"
PUT 1 N$;Z$;A$;Z$;C$;Z$
```

The sector looks like

```
Ronald Reagan, President!The White.....
```

When reading it back, be sure that the STRING "!" has been executed, and BASIC will correctly pause a GET 1 N\$,A\$,C\$ in the desired way.

As an aside, the "!" after C\$, the final string, isn't operationally required. But it is technically correct. If left out, the GET 1 N\$,A\$,C\$ statement will put all characters remaining after the second delimiter, into C\$. If we are using the default string length of 40, then C\$ will be 'Washington, DC 20500' followed by 20 blanks, instead of the 20 character string itself, which was what we intended to do.

NEXT MONTH

Isn't there still a lot of wasted space? YES. Do you have to use delimiters? NO. These, and other exciting tid-bits of info are discussed in next month's newsletter.

.....

BASIC/S & BASIC/Z

by Buzz Rudow

Hopefully, by the time you read this, Systemation's updated MDOS and new CP/M compilers will be available. Since the syntax for both programs are the same, the following list of operational statements shows the power available to you, whether you run MDOS or CP/M. In fact, if I understand correctly, code generated in one system will transfer to the other.

There is an important point to keep in mind. These functions are in addition to all the Micropolis BASIC statements with which you are already familiar. You still have such things as FOR, GOTO, ON GOTO, etc. The new functions mentioned in newsletters 12, 13 and 14 (TAB(x,y), etc.) are still there, and I won't repeat them here.

Either compiler is available through the MUG for \$308 (list \$345). I will fully support both systems. For those of you with an original version of BASIC/S, expect an update from Systemation. The probability is that a media charge will be requested. Paying for the new manual and a disk doesn't seem like much to ask for a revision as extensive as this. It even includes an integrated sort package. However, since there is at least one bug left in the original package, you might have a different opinion. I'll let you discuss those details with Bob Zale.

PROGRAM CONTROL

```
IF x THEN stmt {:stmt} [ELSE stmt {:stmt}]
IF x THEN dest [ELSE dest]
if the expression x is true, the THEN clause (or an implied GOTO) is executed - if false, the ELSE clause (or an implied GOTO) or the next statement is executed
```

DO

statements enclosed in a DO/UNTIL loop are executed at least once, and are repeated as long as the UNTIL expression is false

UNTIL expr

statements enclosed in a DO/WHILE loop are repeated as long as expr evaluates to false

WHILE expr

statements enclosed in a WHILE/WEND loop are executed as long as expr evaluates to true

WEND

terminates a WHILE loop

CONSOLE DISPLAY**BLINK**

output to the console will flash

NOBLINK

output to the console will not flash

RVIDEO

output to the console will be reverse video

NVIDEO

output to the console will be normal video

CLINE

cursor line position of the console

CCOL

cursor column position of the console

ERAEOL

erase from the current cursor position to the end of screen on the console

ERAEOS

erase from the current cursor position to the end of screen on the console

CONSOLE INPUT**INKEY\$(x)**

read x input characters from the console as a string of x bytes, without echoing them to the console

INPUT [slit ;] var {,var} [;]

read a data line from the console, parsing the data into the specified variables - if the last variable is followed by a semicolon, the terminating carriage return is not echoed

INPUT\$(x)

display a field of x '.' and return user input from the console limited to a maximum of x bytes

INSTAT

current console status - true (1) if a key has been pressed or false (0) if not

INCHAR\$(x)

read one input character from the console as a one byte string, without echoing it to the console - if no input is ready, a null string is returned

UPCASE

alphabetic input from the console is forced to upper case

LOWCASE

allow lower case input from the console

PRINTER**PLINE**

cursor line position of the printer

PCOL

cursor column position of the printer

STRING MANIPULATION**MID\$(svar,x) = x\$**

x\$ is inserted into the specified string variable starting at position x

UPCASE\$(x\$)

force all alphabetic characters in x\$ to upper case

LOWCASE\$(x\$)

force all alphabetic characters in x\$ to lower case

DISK OPERATIONS

PUTVEC x [RECORD y] navar, navar
PUTVEC x [RECORD y] savar, savar, x\$
 write a record to random disk file #x, consisting of binary or BCD numeric data from successive elements of the numeric array -or- consisting of string data with the number of bytes from each element determined by the ASCII code of the corresponding byte of x\$

GETVEC x [RECORD y] navar, navar
GETVEC x [RECORD y] savar,savar, x\$
 read a record from random disk file #x, parsing the binary or BCD data into successive elements of the string array with the number of bytes for each element determined by the ASCII code of the corresponding byte of x\$

STATUS(x)

open/closed status of the file associated with file #x - 0 if closed 1 if open

LASTFILE

numeric value equal to the file# of the most recently attempted disk access, whether or not successful

UNFMT

option in CREATE or OPEN to override BASIC pointers, allowing access to all records

OTHER FUNCTIONS

SORT avar, avar, x\$, x
 sort the designated array from the first avar through the second avar, based upon the options x\$ and the key length limit x

INCR nvar

increment a numeric variable

DECR nvar

decrement a numeric variable

DEGREES(x)

convert x (in radians) to degrees

RADIANS(x)

convert x (in degrees) to radians

PI

value of pi to the precision of RSIZE

SPOOL x

print stream is spooled to the sequential disk file associated with file #x

SYSTEM MANIPULATION**PEEKWORD(x)**

read two bytes (low/high format) from memory locations x and x+1

POKEWORD(x) = y

store two bytes y (low/high format) at memory locations x and x+1

PUSH dest

add a destination line number or label to the top of the subroutine stack

POP

delete the last return address from the subroutine stack

SAVE x\$, x, y

save a block of memory from location x to location y as a disk file named x\$

VARADDR(var)

memory location of a variable

.....

COMMERCIAL S/W - NEW FROM DATASMITHA COMPACT FOR CP/M BASIC

SMASH does the same thing to Microsoft Basic that it does to Micropolis. It removes extra spaces and all comments from a non-protected binary format program file. Microsoft requires that line numbers and Basic keywords be terminated by blanks so as not to confuse the interpreter. This also is a way of allowing keywords to be imbedded in long variable names without resulting in any errors. While this is a reasonable restriction for data entry, it is no longer required after the keywords have been converted to tokens and saved in a file. Therefore, the "smashed" programs are impossible to create from the keyboard. One interesting phenomenon is that even though there are absolutely no spaces in the memory image of a "smashed" program, they are re-inserted in the output of a LIST or LLIST command. Apparently the keywords are terminated by spaces when translated from tokens.

A MERGE FOR CP/M BASIC

MERGE is a program which merges one binary format Microsoft Basic program into another binary format program. Unlike its MDOS equivalent, it does not do a multiple merge, but this is easily accomplished with a SUBMIT routine. It offers the advantage of being able to perform the merge without an ASCII file. It also has a feature of being able to delete lines during a merge by the operator simply placing a DELETE statement in the merge file. The DELETE statement is removed along with the specified lines, since it would result in a syntax error if executed. Syntax of the delete statement is:

DELETE STARTLINE-ENDLINE,.....,STARLINE-ENDLINE.

This feature is highly useful in configuring applications software to a particular hardware environment because one merge file makes all necessary patches, including adding and deleting lines.

.....

SOFTWARE REVIEW

by Buzz Rudow

BOOKKEEPING

BOOKKEEPING, written by DATASMITH and available from the MUG for \$189 (list \$250) was designed to do the basic accounting functions for a small to medium sized business. Because the modules are written in Micropolis BASIC, or in Microsoft BASIC-80, and the Ledger titles are user definable, it can also function as a sophisticated home bookkeeping system.

As a bit of background, accounting procedures are designed around the basic premise that ASSETS = LIABILITIES + EQUITY. ASSETS are things like CASH, checking and savings accounts, stocks, and real estate. LIABILITIES are things for which you owe. Included in home LIABILITIES are items such as house mortgage, car loan, and outstanding credit card purchases. For the business, LIABILITIES are Accounts Payable, Taxes Payable, mortgages and other long term debts. EQUITY is the difference of ASSETS and LIABILITIES. For an individual, EQUITY is normally referred to as NET WORTH.

The three major categories of ASSETS, LIABILITIES and EQUITY (or NET WORTH), and their respective sub categories, make up part of a "Chart of Accounts", or "Ledger". The two final major categories of a Ledger are REVENUES and EXPENSES. Business sub-categories under REVENUES are normally different types of SALES. They might be SALARY-1, SALARY-2 and INTEREST for the individual. Business EXPENSES are broken down into Salaries, Raw Material, Purchases, Advertising, etc. An individual may select Auto Expenses, Clothing, Entertainment, Deductable Business Expenses, etc.

The First Step in Setup

For the BOOKKEEPING system, only the major categories of ASSETS, LIABILITIES, EQUITY, REVENUES and EXPENSES are fixed. The number and description of all sub-categories are user defined. Defining these categories, or Chart of Accounts, is the first task a new BOOKKEEPING owner must complete. BOOKKEEPING can be as simple or as detailed as you wish it. Sub-categories can be added or deleted as you go along, so you have the option of starting simple and expanding as you become familiar with the system. Burks has included a complete Chart of Accounts, so you can use or modify his, if you wish.

Lynn's Note

As a rule, I don't go playing with new programs to see how they work - that's Buzz' thing. Although after spending three hours just trying to balance our business checkbook, I was ready for Buzz to tell me about BOOKKEEPING.

Actually, I had read the manual, but being the dumb wife I didn't understand some of the information. Buzz, having the patience of a saint and determination to make me understand, took some time for me and we sat down to run the program. Once I figured out I had to read the prompts first - it kept saying it couldn't decipher my inputs - instead of just hitting any key, things went very well. I even enjoyed it.

The only disappointing thing about BOOKKEEPING, it tells you when you're in the RED. Even so, I think I'll let it balance the checkbook from now on. BOOKKEEPING also shows all the details of what events caused money to come in, and what categories of expenses caused money to go out. So I can see where BOOKKEEPING can be a very useful tool in managing one's finances.

Useful for Business or Home

The point I've tried to make is that BOOKKEEPING is useful in the home. Probably no one will question the usefulness for the business, but one might well question spending \$189 to track home income & expenses.

First, I think people should know the status of their financial position. Most people track these things on file cards or in a notebook. If you are currently tracking your finances, then BOOKKEEPING can save many hours of your time. Error correction is simple, and all account balances are automatically adjusted.

Secondly, the documentation of IRS-allowed deductables is easy, and could well pay for the software package. Set up an account for Sales Tax Expense, Business Related Expense (professional dues, publications, tools, etc.), Investment Expense, and Medical Expense. Though I'm not a lawyer, I'd assume that purchase of the program for use in Income Tax preparation would qualify the price as a deductible expense.

Our transfer of DAMAN's manual bookkeeping to BOOKKEEPING took about four hours. This included learning the system, creating the chart of accounts, and entering the January data. Lynn feels that ten minutes a week is all that's needed to keep it up to date. By the way, all our fussing around with BOOKKEEPING never caused a crash. The error checking is terrific.

.....

MUG LIBRARY UPDATESCP/M ARRIVES

For those of you wishing to try your hand at CP/M, there is now some assistance available through the MUG. The first disk, ambitiously numbered 1001 - to allow plenty of expansion space for the Micropolis operating system software - is a set of 26 games. These are written in Microsoft Basic,

known as MBASIC4 or 5, or BASIC-80.

The games are: HORSRACE, HURKLE, KINEMA, KING, LEM, LETTER, LIFE, LIFE2, LITQUIZ, LOVE, LOVE-P, LUNAR, MADLIB, MASTRMND, MATHDICE, MUGWUMP, NAME, NICOMA, NIM, NUMBER, ONECHECK, ORBIT, POETRY, POKER, PIZZA, and QUBIC.

Disk 1002 is being worked and will contain some CP/M utilities. At this point, it's not complete, but will have the MODEM7, and SUPSUB11 routines. Some of this stuff works directly, some needs configuration. I do not know how to configure in CP/M. You members who want to have the MUG support CP/M should start writing some articles for the newsletter.

The cost for these disks has dropped to \$10 (\$12 outside North America). There is less data on them so I can DISKCOPY under MDOS. (Yes, you can copy CP/M disks under MDOS, and MDOS does it a lot faster.) My CP/M system doesn't presently support both MOD I and MOD II. However, if a MOD II has less than 36 tracks of data, I can DISKCOPY directly to a MOD I. The cost, when a program on your disk is submitted, remains at \$3/\$5.

RESTRUCTURE OF CURRENT MUG DISKS

Within a month or two, I'm going to restructure the MDOS library so that no disk has more than 143K (MOD I capacity). As mentioned above, this allows me to keep only one set of disks. It also allows a more structured library. The present system has all sorts of material on the same disk. My plan is to put utilities together, workable BASIC applications together, BASIC subroutines and programming tips together. The library disk price will come down to \$10/\$12.

I NEED A LITTLE HELP

There has been little help in getting the library programs cleaned up. Perhaps that's because I haven't asked for help, or told you my position on changing current library programs. For the record, a cleaned up program is just as eligible for submission as credit for the \$3 library disk as is a new program. Conversion of MDOS programs to CP/M, the new CP/M games to MDOS, any conversion to BASIC/S or Z, new programs in BASIC/S or Z, or programs in any CP/M language are also legitimate submissions.

One requirement I would like to enforce, however, is to always have the direct cursor codes programmed into a variable in the first lines of the program. That is, if CHAR\$(11) is your clear screen code, write:

```
10 A$=CHAR$(11): ! Set Clear Screen Variable
20 PRINT A$: ! Clear Screen
```

If done this way, members with machines different from yours can easily edit one line to get the program to run for them. I would prefer to have code non-compressed. If people want to CRUNCH or SMASH it, they can. I'd also like you to list the variables used, and their purpose, at the end of the program.

.....

PROBLEMS IN MDOS TO CP/M CONVERSION

Steve Guralnick is converting most of his office software from Micropolis BASIC to CBASIC2 on CP/M. If others are contemplating such a transition, drop me a note. Perhaps we can get Steve to write an article or two on his methods. Using the TR/II-UTL/1 package, he's had no particular problem converting the programs. Currently, however, he's having some problem with data files. I predict, after reading Burks' article this month, that Steve will subsequently have problems with numerical accuracy.

.....

LETTERS

ANOTHER FIX FOR MDOS 'HANGS'

Buzz,
I read your article on how to recover from a "hung" program. One "hang" that even a jump to 4E7 won't fix is if you do an ASSIGN(2,3) when your printer driver thinks a physically disconnected printer is "busy". RES will try to echo every character, even the Sign-On after a jump to 4E7, to the printer, which never takes one. You can fix this one without a reboot by using your monitor to patch D2PORT at 04EB hex. D2PORT is the second of a two-byte sequence that holds the arguments of the last ASSIGN statement executed or the default. Simply change it back to 02 from 03 and then jump to 4E7.

CP/M 'HANG' FIXES

There is a way to restart a Microsoft Basic program too. Since the CP/M TPA begins at 100H, a jump to 100 will usually work. However, Dave Land's Technician, Doug Steen, told me about an even better way. CP/M has a nasty habit of grabbing control of the system after a disk error has occurred in a Basic program. It may be that you just tried to SAVE 3 hours of work and didn't have a diskette in the drive! You can't reload BASIC without destroying your program, which is still in memory. What do you do? Just type SAVE 0 APP.COM and then type APP. You can use any name, of course, but I like APP because it does the same thing APP does under MDOS. When a .COM file's name is typed, CP/M loads it into memory starting at 100H, cleans up the stack, and jumps to it. If the .COM file is of zero length, nothing gets loaded, but the stack still gets cleaned and control is transferred to 100H. Basic will then say "ok" and everything will be as it was.

MICROPOLIS & IBM FORMATS TOGETHER

Have you ever heard of Hawkeye Grafix? They have a number of utility programs for generating graphics using memory-mapped video with the Vector Flashwriter and other cards, a communications program, and a BIOS to access and deblock Micropolis Mod II format for CP/M. The BIOS is also available with additional routines to access a Tarbell 8" disk controller and a Micropolis controller in the same BIOS. Since I have a Tarbell subsystem as well as a Micropolis subsystem, I couldn't pass that one up. It comes in Z-80 source code (Intel Mneumonics) and after only a few hours of patching it for my double-density Tarbell controller, it actually works! I can now run CP/M with drives A and B as Micropolis Mod II and drives C and D as IBM compatible 8" diskettes. This not only gives me a four drive system, but allows me to copy from one format to another.

If anyone is interested in getting 8" single density CP/M software put on Micropolis Mod II or vice-versa, I'll do it for \$25.00, including the destination diskette.

(Hawkeye Grafix is located at 23914 Mobile, Canoga Park CA 91307. Phone (213) 348-7909).

Burks A. Smith
Box 8063, Shawnee Mission, Kansas 66208

.....

MUGSY,

Your discussion of operations on the 'DIR' record (Sept. 1981) was quite helpful. I was able to use your information to write a routine to display only the data files on a disk, their file types and sizes. This is quite helpful in presenting disk contents without other files displayed, and in determining whether the file type must be modified before updating.

If it has any appeal for the newsletter, I'll gladly share it.

In reaction to Burks Smith's article about the

token table (Jan. 1982):

If translation to tokens is done when the program step is first entered, there appears to be no particular operation speed advantage of one position over another. My idea was to group most-used instructions early in the table, if any processing speed was to be gained.

I'd like to change some keywords, or add some alternates, like "L" for "LIST", and "RT\$" for "RIGHT\$". Any cautions?

After your work with BASIC/S, have you any suggestions for a screen dump routine for recording the screen after an error?

Jerry Factor
709 No. Palm Drive
Beverly Hills, California 90210

Jerry: Sure, I'd like to see your data-only DIR. Sounds like a nice touch to have in programs which are asking operators to select a data file. Please send it in.

Burks' article on tokens didn't mean to say that speed was the purpose for the grouping, though early tokens will execute faster than later tokens. The point was that if "IN" is before "INPUT" then the command "INPUT" will decode to "IN" and "PUT", which isn't what you meant.

The cautions for table modification would include looking out for situations like the above - sets of commands where one is contained in the other. You have to have the "long" command first. I assume you realize that if you make any additions or deletions to the table, you won't be compatible with the rest of the world.

If all you do is change the "printout" of a token - say "RT\$" for "RIGHT\$", then there should be no problem. A BASIC program on disk would still have the same token. When your BASIC interpreter sees the token it will display "RT\$", while ours will display "RIGHT\$".

No, I don't have any screen dump routines for BASIC/S at this time. I really haven't done anything with BASIC/S for months. There are a lot of things I want to convert to BASIC/S, but I've been waiting for Rev 1.1, which has turned into 2.0. I use the DDIM statement a great deal and didn't want to recode to do away with it. I also didn't want to have to re-compile a bunch of stuff when the revision arrived. If I had known it was going to be this long before the update, I probably would have worked with it more. But I always thought the revision was "two weeks" away.

Oh well, BASIC/Z is now out and the BASIC/S is "two weeks" away. Then I'll see what I can do with it. If you, or any of the other members, have programs working in BASIC/S or /Z, I hope you consider sending them in for the MUG Library. The MUG will support both languages.

PROBLEMS IN ADVENTURELAND

Buzz,
Is anyone else having a problem running the Adventure program supplied on Disk 3? As soon as 'ADVP2' is chained in and we try to execute the first iteration of statement 7690, my system bombs. BASIC goes off to no-where and must be rebooted before it is usable again. I'm using an IMSAI with 48K of RAM.

Richard C. Rusczyk
6639 12 Terrace N., St. Petersburg FL 33710

LIBRARY RULES STILL FUZZY

Buzz,
I am unclear about what MUG requires of the

programs for trade, and I would appreciate some information. The specific questions I have are:

- (1) Is one program required for each disk?
- (2) Does a program from BYTE count as a submitted program for (1) above?
- (3) Do programs that I've written that are already on a MUG disk count?
- (4) How do I mail disks so that they won't be damaged by the postal service?

Jon Fant, 758D, NMSH,
Univ. of AK., Fairbanks, AK 99701

Jon: Yes, one program is required for each disk.

Yes, programs from Byte (or Kilobaud) count as submitted programs. Other magazines have not been as cooperative. Please indicate if a submission is from a magazine by putting a comment statement to that effect in the first lines of the program. I have to put them on a separate disk.

Yes, you can submit programs that are already on MUG library disks. My current philosophy is that everyone programs differently. Therefore, one can learn a great deal by looking at how two different people accomplish the same task.

The postal service isn't as hard on disks as one might imagine. The Post Office sells padded envelopes for something like a quarter. I've used them and had no problem. Enclosing the disk(s) between two pieces of cardboard, sealing the edges with sturdy mailing tape, and addressing on the cardboard also works.

CLASSIFIED

FOR SALE: Processor Tech "SOL" Computer with SOLOS monitor. S-100 system with 56K RAM. Micropolis MOD II disk drives, controller and software. CP/M and T-MAKER II. Complete System \$2,200 or parts. Call evenings "CST" (608) 788-6677.

Pete Eversole

FOR SALE: UDS-103A-CBS, Bell-103 compatible 300 baud data modem. Answer only, direct connect, with a direct access arrangement (DAA). \$100.

Joe Callaway, 1728 51st St. West
Birmingham AL 35208 - (205) 925-8169
03/01/82

DAMAN	COMPUMAX		
604 Springwood Cir.	MICRO-LEDGER....	\$140	\$140
Huntsville AL 35803	MICRO-PAY.....	140	140
(205) 881-1697	MICRO-REC.....	140	140
=====	MICRO-INV.....	140	140
S/W Price List - 03/01/82	MICRO-PERS.....	140	140
MDOS	ORDER ENTRY....	350	350
MDOS	MAXI-LEDGER....	350	350
=====			

	LIST SALE	DAMAN	
		CATLG I Dsk Cat..	\$ 30 \$ 22
		MAILSYS I Mail..	50 37
		MEMSYS I Gp Mbr.	50 37

<u>ACROPOLIS</u>		<u>DATASMITH</u>	
A-FORTH.....	\$150 \$150	G/L Gnl Ledgr...	\$250 \$189
UTILITY PKG.....	45 45	PAYROLL Payrl...	350 263
		MULMERGE	30 27
BONJOEL		SMASH Rduc BAS..	30 27
DATABASII.....	\$ 50 \$42	SYSLIST pgm.....	30 27
MODFILE-		TEXTCONV BAS....	75 60
MODIMATH D/BASII	50 42	VARLIST Var/ary.	30 27
INVEN-1 Bus Inv.	50 42	Set 5 above.....	150 116
WAMSORT Asm Sort	40 34	DATAMGR	450 450
REACT Reminder..	50 42		
PONY-PICK	300 252		

<u>CHAMELEON SOFTWARE</u>		<u>LENZ SOFTWARE</u>	
GAME 1 Balrog ..	\$ 30 \$ 27	DBANK PER/BUS...	\$ 75 \$ 62
GAME 2 Sisyphus.	30 27		
GAME 3 Mort Fork	30 27	<u>MONK SOFTWARE</u>	
		BEST Bldg Est...	\$295 \$231

ORGANIC SOFTWARE

TEXTWRITER II...	\$125	\$110
------------------	-------	-------

SYSTEMATION

AUTO/EXEC.....	\$ 40	\$ 36
BASIC/S Complr.	345	308
BCOMPARE Compar.	35	31
BEM Exp Module..	65	58
CRUNCH Bas Com..	35	31
DSM-1 80/85 Disa.	65	58
EDIT/S Txt Edtr.	45	40
RUN/S BAS/S pkg.	65	58
SORT/A Sort.....	75	67
TR/II Translto .	55	49
UTL-1 Util Pkg..	95	85
XREF Ref Gen...	85	76

WORDCRAFT

MICRO-LINK.....	\$ 89	\$ 79
-----------------	-------	-------

Prices and availability subject to change without notice.

=====
S/W Price List - 03/01/82
CP/M CP/M CP/M
Micropolis Formatted Disk
=====

LIST SALE

BONJOEL

PONY-PICK.....	\$300	\$252
----------------	-------	-------

CENTA SYSTEMS

CP/M Z80 Microp.	\$189	\$182
------------------	-------	-------

COMPUMAX

MICRO-LEDGER....	\$140	\$140
MICRO-PAY.....	140	140
MICRO-REC.....	140	140
MICRO-INV.....	140	140
MICRO-PERS.....	140	140
ORDER ENTRY.....	350	350
MAXI-LEDGER.....	350	350

COMPUTER MART NJ

CP/M.....	\$180	\$180
-----------	-------	-------

DATASMITH

General Ledger..	\$250	\$189
PAYROLL.....	350	263
SMASH.....	30	27
MERGE.....	30	27

DIGITAL RESEARCH

MAC.....	\$ 90	\$ 90
SID.....	75	75
ZSID.....	100	100
TEX.....	100	100
DESPOOL.....	50	50
PL/1-80.....	500	468
BT-80.....	200	195
CBASIC2.....	150	126
CB 80.....	500	468

INVESTMENT SYSTEMS ANAL

PROPERTY MGMT...	\$725	\$690
PROPERTY ANAL...	250	242
BUS SUPRT S/W...	65	65
SP/LAW.....	125	111

MENDOCINO SOFTWARE

ESTIMATOR.....	\$150	\$150
FRAMING CALU'TR.	150	150
BASIC-AID.....	35	35
BASTRAN.....	35	35
CATALOG.....	35	35

MICRO AP

SELECTOR III....	\$195	\$165
SELECTOR IV....	295	243
SELECTOR V....	495	399
GLECTOR/SELIV...	295	243
GLECTOR/SEL V...	495	400

MICRO CRAFT

LEGAL BILLING...	\$625	\$539
------------------	-------	-------

MICRO PRO

WORDSTAR.....	\$495	\$360
MAILMERGE.....	150	111
WORDSTAR/MM....	645	465
DATA STAR.....	350	295
WORDMASTER....	150	111
SUPERSORT I....	250	176
SPELLSTAR.....	250	176
CALCSTAR.....	295	224

MICRO SOFT

BASIC-80.....	\$350	\$308
BASIC COMPILER..	395	346
FORTRAN 80.....	500	436
COBOL 80.....	750	646
MACRO 80.....	200	182
EDIT 80.....	120	114
MU MATH/MUSIMP..	250	224
MU LISP/MUSTAR..	200	182
M-SORT.....	150	139

ORGANIC SOFTWARE

MILESTONE.....	\$295	253
DATEBOOK II....	295	253
PERSONAL DTBOOK.	150	131
TEXTWRITER 2/3..	125	110

PRO/TEM SOFTWARE

SYNOPSIS.....	\$125	\$119
FOOTNOTE.....	125	119

SORCIM

ACT I 8080/Z80..	\$175	\$161
PASCAL/M.....	395	358
SUPERCALC.....	295	253

SOUTHERN COMPUTERS

RAID.....	\$250	\$241
RAID FP.....	395	372
FPP.....	250	241
RECOVER.....	75	75
RADAR.....	495	463
ISIS.....	250	241

SRX SYSTEMS

CATALOG.....	\$ 75	\$ 75
--------------	-------	-------

SUPERSOFT

DIAGNOSTIC I....	\$ 75	\$ 75
DIAGNOSTIC II...	100	98
FORTH.....	200	182
FORTRAN.....	250	224
RATFOR.....	100	98
SUPER M-LIST....	75	75
TERM.....	150	140
TERM II.....	200	182
TINY PASCAL....	85	85
DISK DOCTOR....	100	98
UTILITIES I.....	60	60
UTILITIES II....	60	60
C COMPILER.....	200	182
STAREEDIT.....	225	203

SYSTEMATION

BASIC/Z Complr..	\$345	\$308
RUN/Z BAS/Z Pkg.	65	58
SORT/B BAS80....	75	67
UNDELETE.....	45	40
UNPROTECT Recv..	70	62

WORDCRAFT

MICRO-LINK.....	\$ 89	\$ 79
-----------------	-------	-------

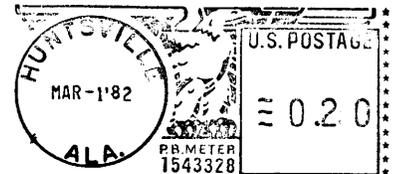
Published Monthly by the MUG
Subscription rates:
U.S., Canada, Mexico; \$18/year; Other, \$25/year

FIRST CLASS MAIL
=====

FIRST CLASS MAIL
=====

MICROPOLIS USERS GROUP

Buzz Rudow, Editor
604 Springwood Circle
Huntsville AL 35803
(205) 881-1697



FIRST CLASS MAIL
=====