MICROPOLIS USERS GROUP

MUG Newsletter #33 - April 1983

Published Monthly by the MUG
Subscription rates:
North America; $18/year
Elsewhere; $25/year - airmailed

**********

## MICROPOLIS SWITCHES TO 1100 SERIES DRIVES

by Buzz Rudow

### 1000 Series Subsystem Production Ends 6/30/83

### Last Chance to Purchase Micropolis Subsystems

As previously reported (MUG 2/83), the Micropolis Corporation has been considering the closing of production for the 1000-series drive. During March, the decision was made - stop production by June 30, 1983. Letters of explanation have been sent to all Original Equipment Manufactures (OEMs) who currently use the 1000 subsystems.

Between now and the end of June, the OEMs will be able to purchase as many 1000-series subsystems, or bare drives, as they wish. After that, the OEMs will need to build their own box and power supply and then use the 1100 series drives.

Micropolis will take orders for these drives until early May. Procurement of parts, and then their assembly, will then take approximately two months.

### Additional Subsystems

Members of the MUG are in a similar situation to the OEMs. If you wnat to get additional subsystems, in either complete or add-on configurations, you need to place an order by May 1st. The MUG/DAMAN has been allowed to act as a Micropolis dealer for this final production run. (See descriptions and prices listed below). Call or write by May 1. Delivery will be made around the end of June.

### Spares?

Do you need to purchase spares? If you are a Vector Graphic owner, the new 1115 drives will work just fine (though there is some question on what REV controller board is needed). Any installation that places the bare drives in a non-Micropolis housing will probably take the 1115 drive without a problem. I tried this on my Blackhawk. All I had to do was place one jumper on the drive, and it works great.

If you're currently running a subsystem, however, the 1115 drive won't fit your enclosure. If, in the future, you have a problem, you'll have to get the 1000 drive repaired, or purchase a reconditioned 1000 drive. There shouldn't be any trouble getting used or reconditiond 1000-series drives, since all the VG people will be replacing the 1015 with an 1115.

In addition, if you're handy at fabrication, you can probably modify the current Micropolis single drive enclosure to have the 1100 drive fit. If you're using a dual-drive subsystem, you probably need to stay with the 1000-series drive. The new drives have control electronics on each drive, rather than the current separate single board that controls two drives.

### Future Repairs

Parts for the 1000 series subsystems will continue to be produced for some number of years (5 to 7). There are several places to get repairs done.

You can do your own repairs with spare parts purchased from the repair shops, and the MUG will also stock some parts, as well as the Maintenance Manuals, alignment and diagnostics disks.

The controller board and the MDOS and Micropolis Basic software will continue to be produced. All these pieces work with the new drives.

Again, if you feel that you have a requirement for a new subsystem or add-on system, you have until May 1st to order. The MUG prices are listed below.

.....

## LAST CHANCE

### MICROPOLIS S-100 SUBSYSTEMS

Available from the MUG/DAMAN until 5/1/83

1053M2 Double-drive, 315 * 2 capacity, List $1534.
Single-head, 630 Kbytes total Daman $ 985.

1053M4 Double-drive, 630 * 2 capacity, List $1888.
Double-head, 1260 Kbytes total Daman $1385.

1043M2 Single-drive, 315 * 1 capacity, List $ 939.
Single-head, 315 Kbytes total Daman $ 685.

1043M4 Single-drive, 630 * 1 capacity, List $1107.
Double-head, 630 Kbytes total Daman $ 840.

The above subsystems are in enclosures, include power supplies, S-100 Controller, cables, manuals, Micropolis MDOS, Assembler, Editor, Basic, & Utilities. Add $14 for shipping of 1053 systems, $7.50 for 1043.

### MICROPOLIS ADD-ON DRIVES

1033M2 Double-drive, 315 * 2 capacity, List $1301.
Single-head, 630 Kbytes total Daman $ 885.

1033M4 Double-drive, 630 * 2 capacity, List $1638.
Double-head, 1260 Kbytes total Daman $1185.

1023M2 Single-drive, 315 * 1 capacity, List $ 698.
Single-head, 315 Kbytes total Daman $ 485.

1023M4 Single-drive, 630 * 1 capacity, List $ 855.
Double-head, 630 Kbytes total Daman $ 635.

Add-on modules are in enclosures, have power supplies, but DO NOT include any of the additional items included with full subsystems. Daisy chain cables can be made by user, or ordered from Daman for an additional $45. Add $14 shipping for 1033 drives, $7.50 for 1023.

.....

## INTERFACING TEXT FORMATTERS

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

One of the most powerful applications of computers is word processing. With appropriate software, generating letters, reports, manuals, and other printed documents is greatly simplified over manual methods, and changing a document is extremely simple. All word processing software consists of two major functions: First, you need a text editor to allow text entry, modification, and file maintenance functions. Second, you need a text formatter to get the document from the computer's memory or disk storage to the printer.

Many word processing systems, such as the popular Wordstar, have combined the text editor and text formatter into an integrated system where the distinction between the two parts is somewhat blurred. However, these two separate functions do exist, whether it is apparent to the operator or not. Another approach, and usually the less expensive one, is to use separate text editing and text formatting software. Most computer systems come with some sort of text editing capability as standard equipment, so only the text formatter must be purchased.

This column was written using Micropolis LINEEDIT and formatted as you see it here with TEXTWRITER from Organic Software. When using CP/M, I use Vector Graphic's SCOPE editor and the CP/M version of TEXTWRITER. I also have WORDSTAR from Micropro and the venerable ED that everyone with CP/M has. With experience using all of the above software, my favorite combination is SCOPE and TEXTWRITER. SCOPE is a very powerful full-screen text editor that is standard equipment on Vector Graphic CP/M that beats the text editing capabilities of WORDSTAR, in my opinion. TEXTWRITER, which is available for both MDOS and CP/M, is one of the few stand-alone text formatters available and has all the features anyone would need.

The actual text formatting function, whether done by a stand-alone program or an integrated system, must be tailored to the type of printer being used. The method of achieving bold-face type, underlining, proportional spacing, and other fancy effects are dependent on the sophististication of the printer itself. Some features, such as superscripts, subscripts and true proportional spacing are simply not possible on most dot-matrix printers, while letter-quality daisywheel type printers can use all these features. For this reason, all word processing and text formatting programs must be configured for the type of equipment being used. Letter quality printers perform advanced functions in response to control codes issued along with the text being sent, so the program must know which printer is being used to send the proper control codes.

For best results with letter quality printers, the text formatting program must be given complete control of the printer, which usually means "patching" machine language codes into the program. Both TEXTWRITER and WORDSTAR come with instructions for doing this along with methods for the selection of different types of printers. When using MDOS and TEXTWRITER, the procedure is as follows:

If you are using a printer that does not have word processing features you don't have to do anything. TEXTWRITER will use the MDOS standard printer output routine and no special features will be available. Unfortuneately, the MDOS printer routines will cause problems if using Diablo or Qume letter quality printers and advanced features like proportional spacing to justify margins. The MDOS printer handler is intended for routine data printout and program listings and has some "features" that are undesirable in this situation. For instance, MDOS counts the number of characters output in a line and automatically issues a carriage return and line feed to the printer whenever the line width set by the ASSIGN statement is exceeded. This allows the listing of lines that exceed your printer's width from a BASIC program, but isn't appropriate for word processing. MDOS also changes a carriage return to a carriage return line feed sequence, changes control-X to a backslash, and changes a backspace to a backarrow.

When printing a line using justified margins and proportional spacing, or if words are underlined or printed in boldface, the program must issue commands along with the text. However, MDOS counts the commands as characters output just as if they were printing characters. This will cause it to issue a carriage return and line feed before the line is actually finished. You can disable this feature by setting the PWRAPFLAG at address 510 Hex to a 1. This will cause MDOS to do simple output without checking to see if page width has been exceeded. This can be easily done with the ENTR command from MDOS, the POKE command from BASIC, or the poke command in TEXTWRITER itself. To re-enable the feature, change address 510 Hex to a zero. Disabling the auto-wrap will help, and the text formatter may work properly, as long as it doesn't issue any backspaces or control-X characters, which still will get special handling.

For complete custom configuration, you will have to follow instructions in the TEXTWRITER manual for using the program's internal printer handler. To do this, you have to know which I/O ports your computer uses for printer status and printer data, and which bit to check to see if the printer is busy or not. As an alternate method, you can patch TEXTWRITER's printer handler routine to a Jump to the MDOS physical printer handler LDOUT, whose address is stored in location 50A Hex in the list jump table. The appropriate code would be LHLD 050AH, PCHL. If you don't know assembly language programming, the instructions would be meaningless to you, so you'll need some help from someone familiar with the inner workings of your computer.

If you know some assembly language programming, patching TEXTWRITER or another word processor for optimum performance with a letter quality printer is well worth the effort.

.....

**********
* *
* VECTOR GRAPHIC SPECIAL INTEREST GROUP *
* *
**********

## VECTOR VENTURES - HARDWARE/SOFTWARE

by Debra Holland
Reprinted from "Vector Views"

The Multi-user systems market clearly demonstrate that the time has come for Vector multi-user microcomputer systems. A system like the 5032, with 5 users and 32MB of disk storage, positions the upper end of the Vector product line in direct competition with the low end of the minicomputer based small business system market.

Multi-user operating systems differ from single user operating systems in several important ways. First, they manage the sharing of the terminals among the system resources, such as the CPU and disk drives. Second, they provide the necessary protection of files and records in the event that two users attempt to access the same file at the same time, and, third, they usually provide background processing functions such as output to printers or communications. In time-shared mutli-user systems, multiple functions are accomplished by allocating time "slices" to each system function and then sequentially "attaching" each user or system task to the CPU for a short period of time. If these time slices are short, each user appears to have the computer to themselves. As additional users are added to the system, new time slices must be created and serviced by the processor. This slows the system response to each user making the system appear rather slow when all users are active. This is particularly true if the tasks require extensive disk accesses or sorts.

In the Vector multi-user systems, the effect of users sharing a Z-80 processor has been partially offset by using the new Z-80B chip which runs at a 6 MHz clock rate. In addition, the Vector Extended CP/M operating system uses such techniques as track buffering and the maintaining of file directories in main memory to further increase the overall speed of the system. The combination of the 5005 or 5032 hardware with the Vector Extended CP/M operating system provides Vector dealers with the most powerful 8 bit multi-user system on the market today. Extended CP/M is compatible with the MP/M (multi-user CP/M) operating system developed by

Digital Research and will run languages and applications programs written for use with MP/M as well as the single user programs written in CP/M.

Other Extended CP/M features include the facility for attaching two printers to the system (one serial and one parallel), the capability for printer spooling and background printing (the print function can be running at the same time as the five terminals are each performing other tasks such as word processing or data entry), and file and record lockout functions (in multi-user environment), just to name a few.

The Vector functional applications packages such as Memorite III and Execuplan II have been modified to operate with Extended CP/M and now have multi-user capabilities. As a result, it is necessary that you specify the Extended CP/M version Memorite III when ordering it for use on systems with the Extended CP/M operating system. The new version of Execuplan II operates with both CP/M and Extended CP/M and has multi-user capabilities.

It is now becoming clear that in the multi-user market there will be no single operating system which achieves the clear dominance which CP/M enjoyed in the single user market. As a result, Vector will be offering additional operating systems for use on the MultiShare systems. The first of these will be OASIS from Phase I systems. OASIS is a well proven and accepted multi-user operating system which supports BASIC and COBOL. OASIS also has facilities for converting CBASIC and MBASIC programs to OASIS BASIC.

**********

**********
*  *
*  BASIC/Z SPECIAL INTEREST GROUP  *
*  *
**********

## BASIC/Z CORNER #3

Steven Guralnick
375 South Mayfair Avenue, Suite 205
Daly City, CA 94015

First, to the mail bag. Dr. H. Daniel Baernstein of Ontario, California, has been installing BASIC/Z on his Compal, using the MDOS version. I have been getting a running account of the process and am pleased to report that he has succeeded. Any of you out there who would like to communicate with him can do so at 524 N. Palm Avenue, ZIP 91762.

I've come up with a cute little routine which I use all the time so here it is. If you want to CREATE a file in BASIC/Z, you need to know if that file is already there and what to do if it is. BASIC/Z does not allow you to CREATE a file with the same name as one already on the same disk. I needed something to handle file creation where the possibility exists that the old version is still there. This is what I came up with:

```
10 CREATE "TESTFILE.FIL" RECLEN 100 ERROR 20:GOTO
   30
20 SCRATCH "TESTFILE.FIL":GOTO 10
30 [PROGRAM CONTINUES HERE]
```

When the program encounters the CREATE instruction on line 10, a successful CREATE at that point will then cause it to execute the GOTO 30 and the program will continue. If it is unsuccessful in doing so, then it will execute the ERROR instruction and jump to line 20. Then the file is scratched and the program returns to line 10 and now the CREATE is successful, so the program then loops around and continues at 30. You can scratch and create a lot of files this way.

I am in the process of converting a massive CBASIC package to BASIC/Z and will be sharing some of the conversion routines with you. I am switching to direct cursor addressing and have worked out a routine which allows me to use SPELLBINDER-written screens for menus, input programs, etc. This capability has made it very easy to: (a) lay out a screen, and (b) find out where everything is because SPELLBINDER will report instantly the location,row and column, of the cursor. So, if you want to reverse video a few words or blink a line, you don't have to write out the whole screen on a piece of paper first. This is what I came up with:

```
70 FORMFEED
80 STRING ""
90 DIM A$(128)
100 OPEN 1 "MENU.FIL" UNFMT RECLEN 128 END 190
110 GET 1 A$
120 FOR I#=1 TO LEN(A$)
130 BYTE$=MID$(A$,I#,1)
140 IF BYTE$=CHAR$(13) THEN PRINT "":GOTO 170
150 IF BYTE$=CHAR$(26) THEN 190
160 PRINT BYTE$;
170 NEXT I#
180 GOTO 110
190 CLOSE
```

It's actually pretty simple. Write the file to disk with SPELLBINDER, but do not, repeat do not, use the "/1" routine. It just complicates things. Then, this is what my routine does. After defining a null for a string delimiter, you DIMension A$ at 128 bytes. Then, open the unformatted SPELLBINDER file and pull out the first 128 bytes in the form of A$. Then, start a loop in which you read every byte of A$, one byte at a time, When that is done, go get another record (at line 170) until you are finished. There are only two checks that have to be made, at lines 140 and 150. At 140 you check for a carriage return and ignore it, except to output a PRINT "", which is the same as a linefeed. The other byte to check for is end of file, 1AH, which SPELLBINDER puts out at the end of the file. When you run into one of those, jump out of the loop and close down.

Otherwise, just print BYTE$ on the screen. It works beautifully. I would like to get it to print a little faster--anyone got any bright ideas? Also, how about one of the WORDSTAR users out there checking this out on that processor and letting me have any changes needed.

We are in receipt of advance word on new enhancements coming in Version 1.03 of BASIC/Z. I am told by the company that by the time you read this, that revision will be available. I will touch on a few things now and then wait until I have a chance to work with the new version and can report more fully on specific routines. What is becoming obvious is that System/Z is trying to put out the best and most powerful BASIC anywhere and those of you who have been thinking about buying it might want to stop thinking and start buying.

Two new implementations which were badly needed are the capability of handling batch compiling and being able to force the debug trace to hardcopy. I write my programs in small modules and use to be able to compile them in batches in CBASIC by using SUBMIT. Having to do it in BASIC/Z one program at a time was a pain and I am delighted that that will end soon. Forcing a trace to hardcopy is also much welcomed--even if you lose a lot of paper, you can sit down somewhere and work your way through the program and the trace and [try to] figure out what is going wrong.

Other enhancements include an INCLUDE command which will allow you to include a library BZS file for compiling with the main package. It will cut down the necessity of having to load in the library file source code into every package where it is needed, especially nice for folks with smaller disk systems.

There is a lot more, including an ALTKEY$ command which will allow you to define a terminator in an input statement. I really need that for the new package I'm working on as I want the user to be able to push ESCape anywhere and bail out. As I

say, when I can actually go hands-on with these new routines, I'll report more.

From time to time, I do get a pet peeve with BASIC/Z and I will vent my spleen here. I don't want anyone to get the idea that these gripes of mine are significant enough to walk away from the program; rather, I throw them out here and if enough of you feel the same way, then I will pass the word on to System/Z for their consideration. This month's peeve is about RENUM. If you attempt to RENUM a program and there is a jump to a non-existent line, what BASIC/Z should do is stop dead. That's the way Micropolis BASIC did it. What it does do is try to renumber what it can and not renumber what it can't. You get lots of good error messages but by that time, the file is in pretty sad shape. So, you'd better SAVE or RESAVE before you do the RENUM routine. 'nuff said.

Jerry Lenz and I have released the Client Ledger Program for general commercial use and would be pleased to put any of you lawyers or your lawyer friends in touch with the program. Modesty compels me to say that it is an excellent package. Completely portable, incidentally, no special cursor routines. It was developed on a Sol and a Sorcerer so you know it's portable.

One of the most powerful sequences in programming is the use of GOSUB:GOTO in the same line. Like so:

```
10 INPUT RESPONSE$
20 IF RESPONSE$<>"1" AND RESPONSE$<>"2" THEN GOSUB
   @ERROR.MESSAGE:GOTO 10
```

Look what you get in two nifty little lines: a user input, followed by an error test, followed by a jump to an error message, followed by a return back to the input, or else onward. Suppose the GOSUB is for every input, good or bad, to see if it is good or bad. Try this:

```
10 INPUT RESPONSE$
20 GOSUB @ERROR.TEST:IF ERRORFLAG=1 THEN GOTO 10
```

After the input is in, you jump to an error checker. If the test says something is wrong, then the test sets the error flag to "1" and returns, otherwise, it just returns. When it returns, you are still on line 20 and whether you back up or go forward depends on whether the error flag was turned on.

Note that the variable input in both cases is called "RESPONSE$". What's nice about it is that an error checker only has to be in the package once because it's only looking to munch a variable called "RESPONSE$". If it checks out o.k., then you convert it to a numeric or some other string further on down the program.

Using labels instead of numbered jumps is one of the best things going in this BASIC. You can MERGE these error handling routines in and never worry about what number you're going to. By the way, speaking of MERGE, I wonder what would be involved to have the BASIC altered to warn you that you are trying to merge a file which is carrying the same line numbers, or some of the same line numbers, as ones already present in the main file.

Keep those cards and letters coming, folks!

**********

## INTRODUCTION TO RAM BANKING FOR STANDARD APPLICATIONS PROGRAMS

by Val R. Krehbiel
Custom Engineering Inc, 3932 W Solano Dr.,
Phoenix, AZ 85019, (602) 973-8904

### HISTORY

Memory Banking, or paging, was originally introduced into large mainframe computers for the purpose of easing memory allocation in applications which required multiple tasks, or operations, to coincide within the same area. The idea was to provide (x) amount of memory for each operation requesting memory space. The algorithms for allocating memory under this method are relatively simple, if a specific page size is pre-defined. This method is now universally used in all computer systems. Although most small computers do not use RAM paging (at least not for this purpose), they do use paging for non-volatile memories, such as disk and tape. An example of this is the extent used by the CP/M operating system, by Digital Research Corporation. Other operating systems often use a more logical type of allocation such as by unit, track, or sector.

In the early days of micro-computers, the small systems were heavily dependent on interpreters, and resident operating systems. This meant that large amounts of RAM were taken up by the operating system and interpreter (often between 20K and 32K). The applications programs were also very often excessively large due to the interpreter code, and it became necessary to extend the memory space beyond the typically limited 64K address boundary. As always, the first solution was to use hardware to achieve this feature. Bankable RAM became an industry by-word, and the ability to use it became system dependent.

Recently the industry has seen an attempt towards memory management. Memory management is the term used for intelligent allocation of memory as described above. Disk systems have used memory management for disk memory since the release of the earliest operating systems, but volatile RAM has unfortunately not been supported in this manner.

### PROBLEM

The 8 bit CPU has an inherent problem in providing a useful memory management system, it is limited to only 64K of addressable memory. 64K is a very decent memory page, but if you consider that the operating system must be resident, then the allocatable memory is quickly reduced. A non-resident operating system is one which would reside in its own area, and would allocate other pages as necessary to incoming tasks. The standard paging system requires that an I/O port be used to output a byte which selects a page of RAM. This is a very difficult method to write programs for, because of the need to save system variables. System variables are those variables which tell the system what is happening, and what needs to happen next. The most common thing which needs to be common to all pages of RAM is the system stack.

There are various techniques used in allowing memory management of RAM. The easiest way to view these techniques is to simply look at what the pages share. For example, the most prevalent systems use a module called the KERNEL. This often consists of a memory management program with a small set of variables. This program and variable set are always addressable by the CPU regardless of the page which has been selected. Other systems share the operating system, and sometimes an interpreter. The variables are then allocated separate pages of RAM to allow user or task independence. The advantage to this method is that less RAM is needed since the only paged RAM is that used for non-common variables. The disadvantage of this method is that user independence is limited, sometimes even to a common program.

While the method of limited shared memory seems to be more acceptable, due to allowing greater flexibility, one must also consider the need for allocating other resources such as disk, I/O, and security.

Most designers have temporarily tried to escape the problems of deciding on an efficient method of resource allocation by again going to updated hardware. The introduction of 16 bit CPU's is often advertised as a solution to higher throughput, but the real problems are still being heavily disguis-

ed. Even the new systems are being released using outdated, and often unworkable, operating systems. The higher throughput is often only a factor of higher system clock rates.

SOLUTION

There has been a lot of talk about UNIX, which is an operating system with resource allocation facilities. Several problems arise with UNIX however. First it is expensive, and does not allow the small computer designer an opportunity to incorporate it in a low cost computer system. Second, it is not available. Despite claims of look alikes and similarities, the designer is usually forced to adapt to someone's special hardware or to software that has many bad features. There is far too much emphasis on compatibility. It is of little value to build a modern computer system with graphics that's compatible with a paper tape adding machine. At the same time design is a very expensive procedure. It is necessary to look objectively at the pros and cons of redesigning anything.

For these aforementioned reasons, the proper design criteria is to use available resources wherever possible. At Custom Engineering Inc. we have attempted to allow uniform integration of existing low cost systems to more advanced software algorithms. The concept revolves around the theory that software tasks are algorithms which are functionally independent. It is not necessary here to devulge the entire structural layout of our design concepts, since we are at this time only interested in presenting the multi-user/multi-tasking program segment. It is sufficient to note that the engineers here at Custom Engineering Inc. believe that many sophisticated functions can be used to upgrade present hardware and software without a significant redesign of the computer.

Memory allocation is the primary purpose of this article, and before we become too far adrift let's look at the hardware setup required to meet the above criteria.

Typical Z80 system configuration:

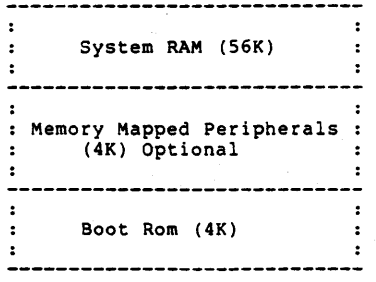



FIGURE 1

Figure 1 provides a view of the memory allocation of a typical dedicated system. The bulk of memory is dedicated to operating system, interpreter or compiler, applications programs, and variables. There is no need to map these areas of RAM separately because they can be varied any way the user desires. Often the system includes ROM and memory mapped peripherals, but this is not important to the description since the designer can optionally alter these also.

In Figure 2 we see a method of allocating memory so that different tasks/users may work simultaneously. In actuality, they do not work simultaneously, but are gaining access to the CPU for a pre-determined slice of time. Note that a non-bankable segment has been allocated to provide system shared variables and program. This is essential for a small computer with limited addressing capability. The advantage of this system is that the CPU spends less time in neutral just idleing. A simple example of this would be if you had a program which printed a long listing. The program printing the listing could operate only if the CRT wasn't being used. This would mean that the user would feel a direct response to his doing some other task, even though the printer was active printing the listing. In short, the user would not have to wait for the printer to finish for him to continue. Another example would be that of having several users operating their terminals concurrently. This system is what is often referred to as a time sharing system.

Since most small computers spend a great deal of time waiting for a peripheral such as a CRT or printer, a great amount of processing power is lost. Response time to the user is often dependant on factors such as peripheral communication speed, and software execution speed. Most users at some time or other wish they could batch a particular operation. A batch operation is simply a low priority time slice operation which acts only when the processor is not busy with another operation.

Time Sharing System for a Z80:

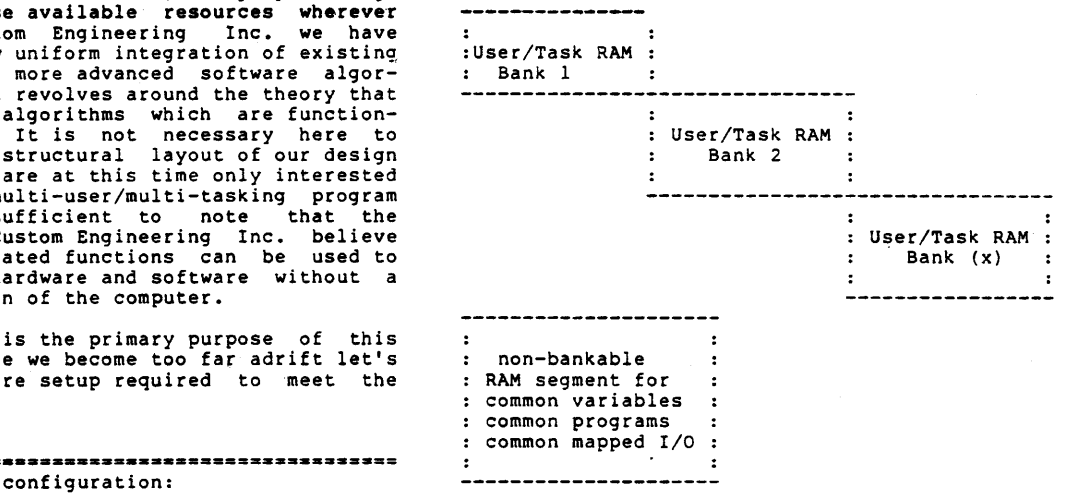



FIGURE 2

In table 1 we see some of the more common problems involved in the use of a memory allocation system. Some of these problems are straight forward, but others are dependant on the system configuration.

| Resource | Allocation Method | Interrupt Method | System Shared | Common Variables |
|---|---|---|---|---|
| Printer | Common | Vectored | Yes | Yes |
| Memory | Bankable | None | No | Messages |
| Disk | Common | On Sector | Yes | All/common |

| Resource | Passing Method | Time Slice | Enable/ Disable |
|---|---|---|---|
| Printer | Strings | 1ms | Yes |
| Memory | Strings | 25ms | Yes |
| Disk | Binary | ? | No |

TABLE 1

As you can see there are a number of questions arising out of each system configuration which will require knowledge of the particular system and the desired configuration of resources. Z80's allow for vectored interrupts which can be daisy chained. This simplifies allocating resources which are interrupt driven, but it is even possible to use a multi-tasking/multi-user program without using interrupts at all.

A timer which interrupts the CPU every milisecond or so is one method of allocating time to a particular task, bank, or user. Another common method

would be for the CRT polling routine to call a counter program for every instance where it is called. This is not as effective as an interrupt driven counter, but will serve for multiple CRT's. Printers which do polling should also call the routine if interrupts are not used. As an added feature, the devices which are inactive need to disable their time slice. This allows for less CPU overhead in dividing time allocation for multiple devices.

The following method is our recommended method for converting to a multi-user/multi-tasking system. All programs are designed to be operating system independent, to allow for the least painful means of expansion.

Package Number 1:
Pager utility package. This program is an enhancement of most operating systems, and allows the programmer to inspect and manipulate multiple banks of RAM. It is very useful for the designer who is attempting to develop a system with multiple banks of RAM.

Package Number 2:
Multi-user package. This program allows the designer to incorporate multiple software packages in the same mainframe. Using separate banks of RAM, he may separately boot or image any area of RAM for a specific task. I/O allocation must be configured by the user, but can most easily be solved by allocating separate I/O for each individual users. More information on this technique is available with the program.

Package Number 3:
Multi-tasking package. This program aleviates some of the problems of allocating interrupt driven I/O. It works only in conjunction with the multi-user package. More documentation is available with the program.

Package Number 4:
Shared Printer package. This program handles multiple printers in a multi-user/multi-tasking environment. The use of multiple printers with spooling to disk or memory is handled without the sacrifice of user dedicated printers.

Package Number 5:
Shared Disk package. This program enhances disk sharing algorithms. This program and successive programs are not operating system independent. Due to the failure of most operating systems to properly structure disk data, it is unwise to be compatible at this point. Conversion programs are available in many instances, but security and expandability of resources are of prime importances at this juncture. This doesn't mean that you must sacrifice data files, but it does mean that you might trash a bunch of useless software. You are moving into a new environment.

Package Number 6:
Custom Engineering Computer Interface Link (CECIL). This is a hardware interface for connecting multiple computers. The purpose of this design will be outlined at a later date. It is sufficient here to point out that multiple processors sharing certain system resources, can provide the type of power not yet seen in the small computer environment of today.

Package Number 7-(x):
Additional enhancements are planned for continued resource management, and high speed program development. For additional information on software and hardware systems offered by Custom Engineering Inc. please contact us. To discuss future enhancements, would at this time take us outside the scope of this article. Sufficient information has been provided to show the direction of development, and that a comprehensive systematic plan does exist.

## RECOMMENDED PROCEDURE

Programs have been written to allow the user and/or designer as much flexibility as possible. It is however, still important to look at the overall picture. It is recommended that an 8K segment be reserved for non-bankable RAM. This segment will provide sufficient working area for small systems utilizing 1-8 users and/or tasks. Each bank should be no larger than 56K, and preferably be located at 0000H to DFFFH or 2000H to FFFFH. Other configurations may be used, but they could be at some point difficult to expand to accommodate more advanced hardware. The non-bankable segment should be viewed as 2 4K segments. At least one segment should be left free for program use. If a segment is needed by the user for shared memory mapped devices, then only 1 4K block should be used.

## CONCLUSIONS

The conversion of existing systems to multi-user/multi-tasking systems, does have its complexities. It is our hope that the work done here at Custom Engineering Inc. can aid in the conversion. We firmly believe in supporting our products, and depending slightly on your hardware, and your operating system, we hope that we can provide a great deal of help with the expansion of your system.

.....

## CUSTOM ENGINEERING SOFTWARE

by Val R. Krehbiel
Custom Engineering Inc., 3932 W. Solano Drive N.
Phoenix, Az. 85019 (602) 973-8904

Our company is now releasing several programs for sale. Since members of our company are members of MUG, and receive its newsletter, we are willing to offer a 20% price reduction to MUG members who buy our programs, as well as offer the majority of our programs for MDOS.

The following represents a list of programs available along with the recommended list price.

| Program | Description | O/S | CPU | Price | Source |
|---|---|---|---|---|---|
| Pager | bankable memory utility | MDOS CP/M | 8080 Z80 Z800 | $35 | Inc. |
| Multi-user | permits multiple users per CPU | MDOS CP/M | Z80 Z800 | $75 | $150 |
| Sedit | screen editor | MDOS CP/M | Z80 Z800 | $125 | N/A |
| IOP | I/O utility | MDOS CP/M | 8080 Z80 Z800 | $35 | N/A |

Other products, soon to be released, include:
* Printer Spooler for multiple banks of RAM
* Shared Disk System for multiple users
* RAM Bankers Package (56k of bankable RAM)
  ( 4k of non-bankable RAM)
  (Multiuser Program)
  (Printer Spooler Program)
  (Shared Disk Program)
* Z8/Z80/Z800/Z8000 Macro Assembler
* SPACE (Structured Programming Algorithms for Computer Expansion) - a highly sophisticated operating system
* Hard Disk Utility for MDOS
* CECIL (Custom Engineering Computer Interface Link) used to link multiple CPU's through sophisticated hardware bus structure
* DART Board (Dual Asynchronous Receiver Transmitter Board) - 8 complete asynchronous channels of I/O (110 to 38.4k baud)
* Low Cost CPU Board (Z80A CPU)
  (64k RAM)
  (2 Asynchronous I/O channels)
  (CTC timer)
  (CECIL interface)
  (CECIL Software Package)
* Hard Disk Sub-System 60-20000MB sub-system for use with CECIL - optional full tape backup with full tape transportability

We are very interested in knowing what the MUG members members think about our software. We are particularly fond of the MDOS, and continue to use and enhanse it. It would be most pleasing if we can help you better utilize your current computer hardware. To discuss your particular interests, or for further information, please call or write.

.....

## CUSTOM ENGINEERING'S PAGER

The pager program is designed to allow the user direct access to multiple banks of RAM. The program is self helping, and a listing is provided with the program. The program comes assembled for E000H and must reside in non-bankable RAM, but may be reassembled for whatever system configuration you may have. The banking scheme used is the most standard in the industry, which is bit mapped for port 40H. This allows 8 possible banks. If other types of banking are done on your system, the program will need to be altered and re-assembled to support any other method. The areas which need to be changed are marked. It is a rather large program, and requires 4K of non-bankable RAM to operate.

Several remarks should be made about possible distructive traps. Bit mapped banks need to have a bit set for a bank to be active. Specifying bank 0 is legal, but could cause the system to disappear. Also banks other than 1,2,4,8,16,32,64,128 are not legal where a bit mapped scheme is used. The program will allow you to enter another number, but if your system uses bit mapped banks, it could actually harm your CPU or memory boards. A 3 for example would enable both banks 1 and 2 on the standard system.

The program was written for an 80 column CRT, and cannot easily be altered. The main place where this is a problem is in the alter byte command which will allow the user to generate 5 entries per line. If you do not have an 80 column terminal, then use CR or Control J to start a new line as needed. No special terminal characters are used.

Also be cautious of the execute or goto command. In order for the program to actually switch banks and execute some address, it must first allocate a new stack area in non-bankable RAM. If you do not properly reinitialize the stack, you may not be able to return. Also if you do re-enter the pager program after running in another bank, don't attempt to specify an originating bank other than the one you came from unless the stack is in non-bankable RAM. If the original stack disappears, you probably will too.

You may have to reassemble this program for a different address before you can even use it, but many different memory configurations are available for bankable RAM, and CE can't possibly know everyone's needs. In many cases it should be ready to run as is, but some will need to reassemble for a different execution address. Still others will need to change the port number from 40H to something else. In rare cases there will be a need to change the routines that select each bank.

It is rather unusual for a company to provide a list for any program. However, Custom Engineering has decided to allow the listing to be released free of charge. In return, CE would appreciate knowing what banking scheme you are using, as well as any other changes you might make to the program.

.....

## AUTOMATIC DEBUG, SCRATCH, EDIT and ASSEMBLY

by Aaron Binder
P. O. Box 90, Pattaya City
Cholburi, Thailand

I find Carl J. Singer's program "SUBMIT" (MUG #29-3) very helpful when creating, revising, or debugging assembly language programs. There are 8 commands which have to be done over and over again. With submit the process becomes automatic.

The transfer editor file, which I call "SEA" for scratch, edit, assemble is:

```
80 LOAD "PROG.O"
90 DEBUG-68
100 TYPE "PROG.O" 0
110 SCRATCH "PROG.O"
120 LINEEDIT
130 ASSM "PROG.A" "PROG.O" "CE"
131 ASSM "PROGP.A" "PROG.O" "P"
133 TYPE "PROG.O" 17
135 PROG.O
```

To invoke enter >SUBMIT "SEA"

After PROG.A is created in the editor, SUBMIT "SEA" is invoked every time a revision is to be made or debug is desired. The only entries that are made outside of SUBMIT, are the debug exit EXEC 4E7 or the lineedit command LOAD PROG.A or the lineedit exit DOS. If at any point printout of the assembled version is desired, RENAME PROG.A to PROGP.A.

.....

## -DISK.FILE

by Julius C. Martin
Greenwood, WV 26360

My DISK CATALOG, which is on MUG Library Disk 24, solved one of the problems everyone has in keeping track of which program is on which disk. The current version requires a video mapped system and doesn' provide any information about the file type. However, it works and provides good hardcopy records on an Epson printer. Designed as it was to make the computer do most of the work, the disk numbering scheme used was versatile and workable. However, I didn't like having to allocate a full track just for the sake of the numbering system.

To make that track do double duty, I offer my "-DISK.FILE" program. It is based on Rudow's "TESTDIR" program in MUG #14. By filling an array with the ASCII names of the files in the DIR, we can provide an automatic menu to list and run any program selected. The program is in BASIC and only BASIC programs are placed in the menu. All filenames on the disk are displayed.

The program listing below has been spread out to make it readable. When entering it, pack each line as indicated and leave out all spaces not in quotes. To use the program, the file type of the disk number files must be changed from 00 to 10 so BASIC can save the menu program in that space. If the disk is a new one, skip that step. Ordinarily, it is a good idea to put the disk number at the start of the DIR to speed up the DISKCAT program. The program should be entered and saved under the name "-DISK.FILE". Then SAVE it on each disk as "-DISK.###" where ### is the number of the disk in question. For a new disk, it is saved as a new BASIC file, numbering the disk and entering the auto-menu program at the same time. To use the program, PLOADG "-DISK.###" where the ### is the disk number (which should also be on your disk label).

The program itself is straight forward. Line 40 allows only BASIC programs, decimal types 16-19, to be selected for the menu. Line 50 builds the filename into an array slot. The CHAR$(12) in Line 10 clears the screen on the EXIDY Sorcerer.

```
10 PRINT CHAR$(12);
   :PRINT TAB(21)"Martin Auto-Menu"
   :PRINT
   :DISPLAY "DIR"
   :PRINT
   :Z$="Z9"
   :M%=-1
   :E%=0
```

```
    :DIM A$(77,10),B$(250)
 20 OPEN 1 "DIR" ERROR 120
    :EOF(1)=16
    :FOR L% = 3 TO 12
    :GET 1 RECORD L% B$
    :J% = 16R3B7
 30 FOR K% = 1 TO 128 STEP 16
    :IF PEEK(J%+K%) = 255 THEN 60
 40 IF PEEK(J%+(K%+11)) < 16 OR PEEK(J%+K%+11))
      > 19 THEN 60
 50 M% = M% + 1
    :FOR I% = 0 TO 9
    :A$(M%) = A$(M%) + CHAR$(PEEK(J%+K%+I%)))
    :NEXT I%
 60 NEXT K%
    :NEXT L%
 70 EOF(1) = 1
    :CLOSE 1
    :IF E% = 1 THEN END
 80 PRINT TAB(24)"Menu"
    :FOR I% = 0 TO M% STEP 3
    :PRINT FMT(I%,Z$);"  ";A$(I%);
       TAB(17) FMT(I%+1,Z$);"  ";A$(I%+1);
       TAB(33) FMT(I%+2,Z$);"  ";A$(I%+2)
    :NEXT I%
    :PRINT FMT(M%+1,Z$);"  END PROGRAM"
 90 PRINT TAB(5);
    :INPUT "Enter number of program to run";A%
    :IF A% < 0 OR A% > M%+1 THEN 90
100 IF A% = M%+1 THEN END
110 PLOADG A$(A%)
120 PRINT "ERROR ";ERR$
    :E% = 1
    :GOTO 70
```

As with any program which affects the DIR, be sure you have a backup copy of the disk in use. Double check your entry of the listing before you run the program. Try it out on a disk you can afford to lose to be sure the program is working right. I would appreciate learning of any improvements. Perhaps a machine language version?

.....

## LETTERS

Buzz: On the subject of reduced prices, I seem to have some relevant data. In December, I bought Nevada Edit and Nevada Cobol from Chuck Ellis, and my sales receipt was number 4845. I immediately sent off for Nevada Fortran, and a month later I received it. The sales receipt was number 6838. So, it appears that Ellis Computing wrote about 2000 sales slips in a 30 day period. That would be a minimum gross of around $60,000 (assuming a minimum 29.95 plus $4 shipping plus $1.80 sales tax per order).

Robert Brown, 1731 Elroy Dr, Lemon Grove, CA 92045

Bob: Sounds good for Mr. Ellis. Wonder why I only sold the COBOL. I still have Nevada Edit and Nevada Pilot. Doesn't anyone want to rid my inventory of these? $29.95 plus $3 shipping.

.....

Buzz: It might be of interest to some of the MUG readers to know of a price worthy source of disks. I have been purchasing BASF 100% certified disks at a price of $1.85 each in lots of one hundred from an outfit in New Jersey. Address is as follows:

Data Exchange, Inc., 280 Dukes Parkway, P.O. Box 85, Somerville, NJ 08875. Phone (201) 725-6680.

Sig Busch, CMR Box 6591, APO NY NY 09633

.....

Buzz: I would like to bring a "bug" in one of your programs to your attention. It is in the Disk Catalog program on MUG Library Disk #10.

Line 4120 of the "DIRSORT" program should be changed to read:

```
FOR Q%=1 TO Q%-1
```

The line presently causes the first record of the sort file to be less than the 247 character length of all other records in the file. This caused some problems for me in the printout of the sorted file.

G. D. Hershell

.....

Buzz: Does DAMAN have a copy of JRT Pascal 3.0? On Micropolis? Of the packages I am looking at (cheap, of course), that is the one I want most, and is the one that doesn't come from the company on Mp format.

I will be sending some programs to the MUG library soon. And speaking of which, I have already contributed some stuff and it hasn't shown up in the library???

Marc Lewis

Marc: No, I don't have a JRT Pascal 3.0. However, if you want that, or any other disk, converted from 8" to Micropolis (MOD I or II) (or visa-versa), I'll do so for $15. That includes the cost of the destination disk and return postage.

Your are definitely correct about the Library Disks being behind. I got caught up in trying to segment them to specific categories of use. That meant dumping, and trying to understand the purpose of each program. Some have adequate documentation, some don't. Anyway, I'm trying to just get the programs out on the 900 and 1900 series of disks. Hopefully, this will all be up to date soon.

.....

## CLASSIFIED

FOR SALE: Vector Graphic S-100 boards (guarenteed to run): 8" FDC, $175. 48K RAM, $149. 64K RAM, $200. Mindless Terminal with Flashwriter II board, $950. Peachtree Software (GL, AP, AR, PR, INV), $300. C-Basic, $50. MDOS with BASIC, $40. Does not work: ZCB single board computer, $100. Z80 board, $60.

David Paden, 5737 11th Avenue South, Birmingham AL 35222, (205) 595-6792 (leave message).

.....

FOR SALE: Vector Graphic System B, 56K, 2 MOD II Micropolis drives, MDOS, CP/M 2.1, Electric Pencil, Memorite II, Basic/S, Bitstreamer II, all documentation, 137 5" disks. $1995

Orville Fudpucker, (214) 423-6214

.....

FOR SALE: Vector Graphic components. 3 Micropolis MOD II drives with controller, Flashwriter and Mindless Terminal, PROM-RAM board, 56K RAM, IMSAI SIO-2, IMSAI mainframe, Cromemco Z80 ZPU, Hayes S-100 modem, cabinet and software. Make offer on all or part.

Roger Pogue, after 6PM PST, (415) 731-2417; or (415) 542-8394 during working hours.

.....

FOR SALE: Two Micropolis MOD 1022-I drives (one is a 1041-I that has been upgraded with its own power supply from Micropolis), controller board, user's manual (3.0 and 4.0) and the Maintenance manual, 50 (at least) 5" disks, some with programs. This is a distress sale. I am asking $350 for all, and the price is firm. Buyer to pay shipping costs.

Paul Kittle, P.O. Box 1285, Loma Linda CA 92354. (714) 796-1300 and leave message, or (714) 824-4585 8AM-1PM California time.

.....

DAMAN, Suite 14, 3322 S. Memorial Pkwy., Huntsville AL 35801 - Phone (205) 883-8113

**ACCOUNTING PACKAGES**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | BOOKKEEPING (DS) | 250 | 240 | 218 | 188 | [ ] | PAYROLL (DS) | 350 | 336 | 305 | 263 |
| [ ] | MICRO-LEDGER (CM) | 140 | 173 | 157 | 136 | [ ] | MICRO PER'NL (CM) | 140 | 173 | 157 | 136 |
| [ ] | MICRO-A/R (CM) | 140 | 173 | 157 | 136 | [ ] | MAXI-LEDGER (CM) | 350 | 430 | 390 | 337 |
| [ ] | MICRO-A/P (CM) | 140 | 173 | 157 | 136 | [ ] | ORDER ENTRY (CM) | 350 | 430 | 390 | 337 |
| [ ] | MICRO-INV'Y (CM) | 140 | 173 | 157 | 136 | [ ] | INVENTORY ONE (BJ) | 50 | 50 | 46 | 40 |
| [ ] | DISK-BANKING | 75 | 74 | 67 | 58 | | | | | | |

**COMMUNICATION PACKAGES**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | MICRO-LINK | 89 | 89 | 80 | 69 | | | | | | |

**DATA BASE MANAGERS**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | DATABASE TWO (BJ) | 50 | 50 | 46 | 40 | [ ] | MODIFILE/MATH (BJ) | 50 | 50 | 46 | 40 |
| [ ] | DATA MANAGER (DS) | 450 | 432 | 392 | 338 | [ ] | REACT (BJ) | 50 | 50 | 46 | 40 |

**GAMES, MDOS**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | BALROG | 40 | -- | 40 | --- | [ ] | SISYPHUS | 40 | -- | 40 | --- |
| [ ] | MORTON'S FORK | 40 | -- | 40 | --- | [ ] | GAME DISK | 35 | -- | 33 | --- |

**LANGUAGES**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | BASIC/Z+1 YR MUG | 345 | 331 | 300 | --- | | | | | | |

**MICROPOLIS**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | MDOS VER. 4.0 | 75 | 85 | 78 | 68 | [ ] | DISK MAINT'CE MANL | 50 | 61 | 56 | 49 |
| [ ] | ALIGNMENT DISK/SS | 50 | 53 | 48 | 42 | [ ] | ALIGNMENT DISK/DS | 100 | 101 | 92 | 80 |
| [ ] | DIAGNOSTIC DISK | 50 | 50 | 46 | 40 | [ ] | H/W MOD (SHAW) | 20 | 24 | 22 | 20 |
| [ ] | 1115 BARE DRIVE (REPLACES 1015) | | | | 316 | [ ] | DRIVE MOTOR 100012-01-2 | | | | 36 |
| [ ] | DRIVE BELT 725-1201-5 | | | | 7 | [ ] | HEAD PADS (2) | | | | 2 |
| [ ] | SAUNDERS MAGNALUBE (GREASE) | | | | 5 | [ ] | 16-SECTOR DISKS, SSDD, BOX OF 10 | | | | 37 |

**MICROPOLIS USERS GROUP**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | YEAR 1 BACK ISSUES | 18 | -- | -- | 18 | [ ] | YEAR 2 BACK ISSUES | 18 | -- | -- | 18 |
| [ ] | YEAR 3A BACK ISS'S | 9 | -- | -- | 9 | | | | | | |

**UTILITIES, BASIC**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | PSORT | 30 | 31 | 28 | 25 | [ ] | WAMSORT | 50 | 50 | 46 | 40 |
| [ ] | MULTIPLE MERGE (DS) | 30 | 29 | 26 | 23 | [ ] | SMASH (DS) | 30 | 29 | 26 | 23 |
| [ ] | VARIABLE LISTR (DS) | 30 | 29 | 26 | 23 | [ ] | BASIC UTIL'S (GS) | 50 | 50 | 46 | 40 |
| [ ] | BASIC COMPARE (BZ) | 35 | 34 | 30 | 26 | [ ] | BASIC EXPANS'N (BZ) | 65 | 62 | 57 | 49 |
| [ ] | CRUNCH (BZ) | 35 | 34 | 30 | 26 | [ ] | SORT/A (BZ) | 75 | 72 | 65 | 56 |
| [ ] | CROSS REFER'CE (BZ) | 85 | 82 | 74 | 64 | | | | | | |

**UTILITIES, MDOS**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | MDOS EXTENTIONS | 50 | 50 | 46 | 40 | [ ] | TEXT CONV'R (DS) | 75 | 72 | 65 | 56 |
| [ ] | UTILITY SET (DS) | 150 | 144 | 131 | 113 | [ ] | SHAW UTIL'S (SL) | 45 | 45 | 41 | 38 |
| [ ] | AUTO/EXECUTE (BZ) | 40 | 38 | 35 | 30 | [ ] | DISASSEMBLER (BZ) | 65 | 62 | 57 | 49 |
| [ ] | TRANSLATOR II (BZ) | 55 | 53 | 48 | 41 | [ ] | UTILITIES I (BZ) | 95 | 92 | 83 | 71 |
| [ ] | FILE UTILITY (GS) | 50 | 50 | 46 | 40 | [ ] | SYSTEM LIS'R (DS) | 30 | 29 | 26 | 23 |

**WORDPROCESSORS**

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | EDIT/S (BZ) | 45 | 43 | 39 | 34 | [ ] | TEXTWRITER II (OS) | 125 | 130 | 119 | 103 |

PLANS: A - Phone support, exchange privilege, 90 days. MDOS - 04/01/83
B - Phone support, exchange privilege, 30 days.
C - Support limited to supplied documentation, no exchange except for bad disk replacement.

TERMS: Prices include cash discount. Add 4% for charge or COD orders. VISA and Master Card Accepted. Shipping to North America is included in all prices. Prices subject to change without notice.

CP/M SOFTWARE

## ACCOUNTING PACKAGES

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | BOOKKEEPING | 300 | 288 | 261 | 225 | [ ] | PAYROLL | 400 | 384 | 348 | 300 |

## DATA BASE MANAGERS

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | CONDOR 1 | 295 | 290 | 262 | 226 | [ ] | CONDOR 3 | 650 | 632 | 573 | 494 |
| [ ] | CONDOR 1 ENHANS | 405 | 394 | 357 | 323 | [ ] | dBASE II | 700 | 680 | 616 | 531 |

## FINANCIAL SPREADSHEETS

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | SUPERCALC | 295 | 291 | 264 | 228 | [ ] | SCRATCH PAD | 295 | 290 | 262 | 226 |
| [ ] | PLAN 80 | 295 | 291 | 264 | 228 | [ ] | VISI-CALC | 250 | 264 | 239 | 206 |

## LANGUAGES

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | ADA (SuperSoft) | 300 | 294 | 267 | 230 | [ ] | BASIC-80(5") | 350 | 371 | 325 | 290 |
| [ ] | BASCOM | 395 | 418 | 378 | 326 | [ ] | BASIC/Z | 345 | 331 | 300 | --- |
| [ ] | C-BASIC | 150 | 141 | 128 | 110 | [ ] | CB 80 | 500 | 568 | 515 | 444 |
| [ ] | C (SuperSoft) | 250 | 246 | 223 | 193 | [ ] | T-PASCAL (Super S) | 85 | 85 | 77 | 66 |
| [ ] | FORTRAN (Super S) | 425 | 416 | 377 | 325 | [ ] | FORTH (Super Soft) | 200 | 197 | 178 | 154 |

## UTILITIES

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | ACTIVE TRACE | 125 | 125 | 113 | 98 | [ ] | CATALOG | 75 | 75 | 68 | 59 |
| [ ] | COMPRESS | 60 | 61 | 55 | 47 | [ ] | DIAGNOSTICS | 125 | 125 | 113 | 98 |
| [ ] | DISK DOCTOR | 100 | 101 | 91 | 79 | [ ] | FILE FIX | 100 | 99 | 90 | 78 |
| [ ] | POWER | 149 | 146 | 133 | 114 | [ ] | DISK EDIT | 100 | 101 | 91 | 79 |
| [ ] | UTILITIES I | 60 | 61 | 55 | 48 | [ ] | UTILITIES II | 60 | 61 | 55 | 48 |
| [ ] | ULTRASORT | 150 | 155 | 141 | 121 | [ ] | SUPER SORT II | 250 | 245 | 222 | 191 |
| [ ] | SUPERVYZ | 150 | 123 | 112 | 96 | [ ] | CP+ | 150 | 146 | 132 | 114 |

## WORDPROCESSORS

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | SPELLBINDER | 495 | 368 | 334 | 288 | [ ] | WORD STAR | 495 | 483 | 438 | 378 |
| [ ] | WORD STAR/MAILM | 645 | 630 | 571 | 493 | [ ] | WORD S/MM/ST | 845 | 822 | 745 | 643 |

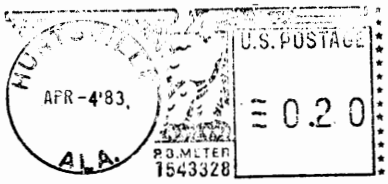
## WORD PROCESSOR UTILITIES

| | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C | | PROGRAM | LIST | PLAN-A | PLAN-B | PLAN-C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ ] | GRAMMATIK | 75 | 75 | 68 | 59 | [ ] | PROOFREADER | 50 | 51 | 46 | 40 |
| [ ] | SYNOPSIS | 89 | 89 | 80 | 69 | [ ] | SPELLSTAR | 250 | 243 | 220 | 190 |
| [ ] | MAIL MERGE | 250 | 243 | 220 | 190 | [ ] | BIBLIOGRAPHY | 125 | 125 | 113 | 98 |

FIRST CLASS MAIL

FIRST CLASS MAIL

DAMAN
Suite 14, 3322 S. Memorial Pkwy.
Huntsville AL 35801
(205) 883-8113




FIRST CLASS MAIL