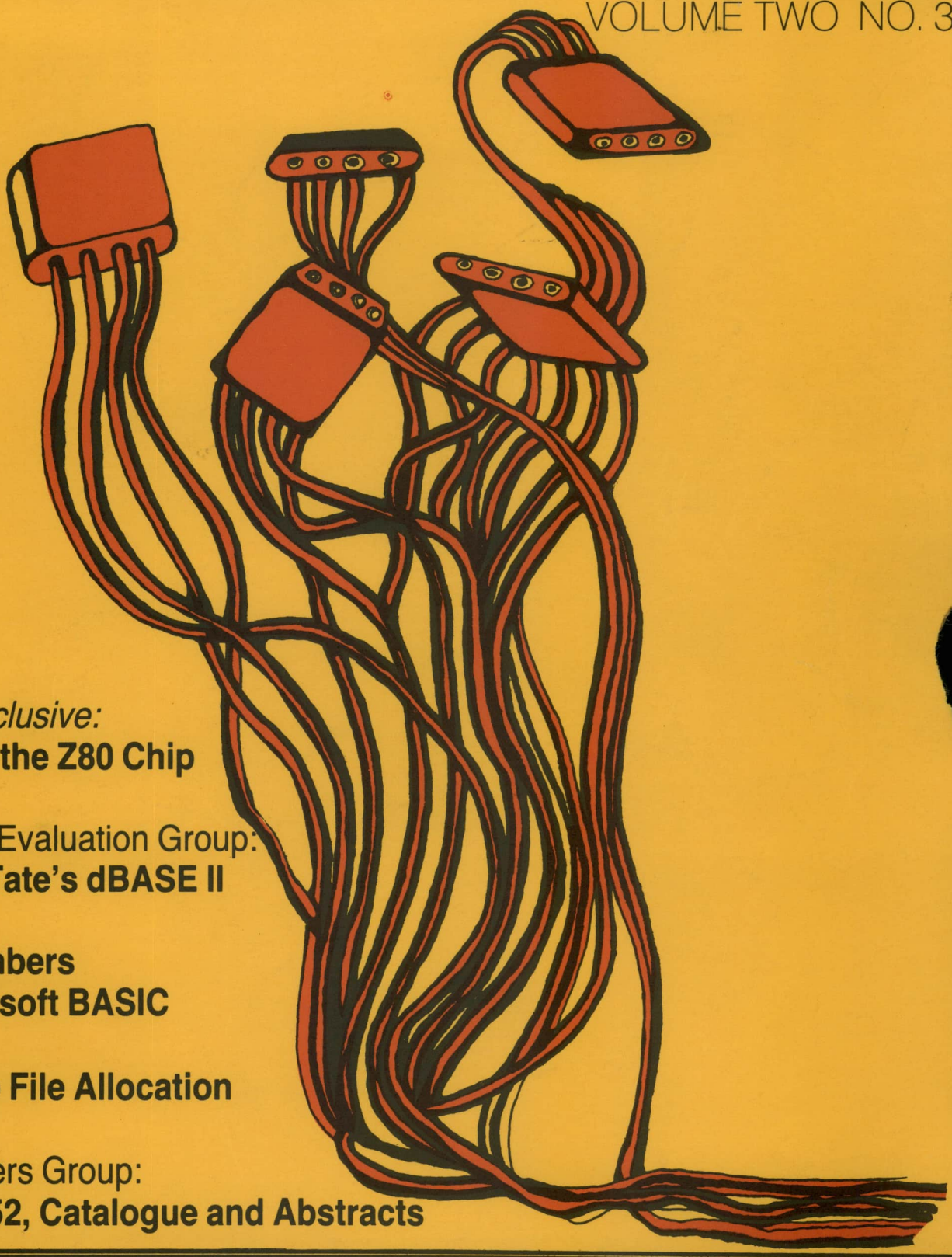
AUGUST 1981

LIFELINES®

$2.50

VOLUME TWO NO. 3



Lifelines Exclusive:
A Bug in the Z80 Chip

The Software Evaluation Group:
Ashton-Tate's dBASE II

Random Numbers for Microsoft BASIC

CP/M Bit Map File Allocation

The CP/M Users Group:
Volume 52, Catalogue and Abstracts

LIFELINES®

Editor-in-Chief: Harris Landgarten
Managing Editor: Jane V. Mellin
Production Assistant: K. Gartner
Administrative Assistant: Susan M. Sawyer

Volume II No.3 August

CONTENTS

Lifelines, Volume II, Number 3. Published monthly. The single copy price is $2.50 domestically, including the U.S., Canada, and Mexico. The single issue price for copies sent to all other countries is $3.60. A one year's (12 issues) subscription is priced at $18.00, when destined for the U.S., Canada, or Mexico, $40.00 when destined for any other country. All checks should be made payable to Lifelines Publishing Corporation, Foreign checks must be sent in U.S. dollars, drawn on a U.S. bank; checks, money orders, VISA, and MasterCard are acceptable. All orders must be prepaid. Lifelines is published by Lifelines Publishing Corporation, 1651 Third Avenue, New York, N.Y. 10028; telephone: 212-722-1700. Please send all correspondence to the Publisher at the above address. Postmaster send change of address to the above address. Application to mail at second class postage pending at New York, N.Y.

Editorial Comments

The final weeks of June, while the rest of the country was preparing for vacation, were busy for the microcomputer industry. The spring COMDEX show was held in New York from June 23rd through June 25th. During the short history of the COMDEX show there have been a surprising number of disastrous incidents. The now infamous MGM Grand fire marred last fall's show in Las Vegas. This spring's show was originally scheduled in conflict with a competitive event in Atlantic City, a fact which disrupted exhibitor plans and made some skip COMDEX completely, even though the Atlantic City show was later cancelled under questionable circumstances. Those who planned to attend were treated to a threatened air controller strike, scheduled for the morning of the 22nd. Despite the settlement of that strike in usual labor negotiation fashion at 11:59 PM on the 21st, it seems that the fear of being stranded in New York City proved too much for many. Whatever the reason, the show was populated by many exhibitors (who were already committed) and by very few paying customers. It is my feeling however, that the unhurried and uncongested atmosphere helped to promote business between the serious computer folks attending. The fact that attendance was sparse did not prevent some important product announcements. I was particularly impressed by three new products, two of which were introduced by Lifeboat Associates.

Microsoft helped to bring CP/M to the consumer computer market last year with the Z80 SoftCard for the Apple. Now Lifeboat Associates has introduced the SoftBox for the Commodore PET. The SoftBox is a self-enclosed chassis which contains a Z80 CPU and 64K of memory and connects to the PET via the IEEE port. Installation merely involves connecting the cables between the PET, SoftBox, and disk drives. A PET can then run a full 64K operating system. The package will retail for about $900 and will include the operating system. Lifeboat also announced CP/M compatibility for the TRS-80 Model III. This includes a small hardware board which plugs into the Z80 socket within the Model III, enabling the machine to run a standard 64K operating system. Prices and delivery dates are not yet known, but it is rumored that there will be a version for the Model I as well. With CP/M compatibility on Apples, PETs and TRS-80s, Atari cannot be too far behind.

The third new product interested me so much that I bought one. It's called Type and Talk and it is made by Votrax. This peripheral connects to the computer via an RS232 port, and enables the computer to synthesize speech with no internal software support. You simply send ASCII strings of English text to the Type and Talk, and it speaks. All pronunciation software is contained within the unit itself, allowing for an unlimited vocabulary with no internal memory overhead. Direct access to the phoneme synthesizer is allowed for words which cannot be reproduced through other means. For those who do not have an extra RS232, the unit can be daisy chained with your terminal or printer with some fancy cabling. This unit makes it so easy to add speech to programs that its use will be widespread, providing software authors with the incentive to add speech support to their programs. The Type and Talk unit retails for $350.00 and requires a user-supplied cable and speaker. My unit has worked flawlessly (it has an Irish accent) and has added a new dimension to my custom programs. Those who skipped this show missed a winner.

The 5.3 version of the Microsoft BASIC Compiler has arrived, and among other things, it contains CHAIN with COMMON. Perhaps more significantly, a large portion of the library (15.5K) has been separated into a BRUN runtime module. BRUN contains routines that are almost always used in large programs, so little or no memory overhead is forfeited in major applications. Furthermore, the execution speed of programs is not affected. BASLIB is still searched for those routines which are not in BRUN. An executable program which requires BRUN will automatically load it from the default disk upon initialization, making it mandatory for distributed packages. The effect of this is several fold: L80 can link larger programs since it does not have to deal with a major part of the library; large amounts of disk space are saved since BRUN must only appear once on the disk instead of being imbedded in every COM file; Microsoft can sell BRUNs to software consumers, solving the royalty dilemma in much the same manner that I suggested in my column several months ago. It is not clear as yet in what manner the BRUN package will be distributed and at what price. Next month I will have more definite information. This could be the key to popularizing the BASIC Compiler among professional software authors. A second version of the library call OBSLIB is supplied for those applications where BRUN is inappropriate, such as ROM-targeted programs. In this case, the compiler works in much the same way as the old version, with the exception that a FORTRAN-like COMMON statement is available for passing variables to FORTRAN subroutines or assembly language programs. CHAIN with COMMON is not implemented if the OBSLIB library is used. PLINK-II does not work with the new compiler in either mode of operation. Corrections to PLINK-II will be made.

In the last few months, I have chastised Microsoft for failing to produce a BASIC Compiler with the CHAIN with COMMON feature by promised dates. Perhaps in commenting on this isolated issue, I have caused some readers to view my thoughts on Microsoft out of perspective. I feel that Microsoft is the most important software producer in our industry, and that without the contributions of Bill Gates and Paul Allen, the microcomputer's rapid acceptance would not have been possible. If the BASIC Compiler were not an important product, no one would care what features it possessed. As a programmer, it is apparent to me how unpredictable time tables for computer programs can be. I feel that one of the major services that Lifelines supplies to our readers, many of whom depend upon Microsoft for their living, is timely and hopefully accurate information

upon which to base decisions. My comments on the status of the BASIC Compiler were made in this vein and no personal attack was intended. It is my hope, that in the future Microsoft will help in correcting and preventing any inaccuracies about their products by communicating with Lifelines more freely.

Harris Landgarten

Letter to the Editor

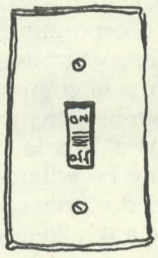
Dear Harris Landgarten,
I found your article "A Review of PLINK II", very interesting and have since placed an order for it. But, I presently have the FORTRAN 80 and Link-80 which comes with it, but I do not find sufficient instructions in the manual to use the overlay capability which your article indicated it has. I do find reference to P and D commands and to overlay error messages, but I am not swift enough to piece together the missing links. I have a Vector Mz and am enjoying transferring programs I have developed for large I.B.M. systems into the microcomputer. I do not know how to talk about memory maps or how to find my way around memory locations, how to "jump" to or "put" or where to overlay to. If you could (a) write an article along these lines or (b) steer me to a reasonable intro discussion or (c) explain how to overlay with Link-80, or all of the above, I will be very appreciative.
Thank you,

Jack Byrne, Ph.D.
Psychologist
Honolulu, Hawaii

Reply

Dear Dr. Byrne,
It appears that my review of PLINK-II lost you quite badly. No true compiler, such as FORTRAN-80, has any inherent overlay ability in and of itself. It is the purpose of the FORTRAN-80 compiler to translate FORTRAN source code into machine code in Microsoft's relocatable format. The compiler itself has no idea of where or how the program will occupy the computer's memory. The linker takes the compiler output and assigns it to specific memory location before outputting an executable program. The reason you have found no reference to overlays in the Link-80 documentation is because Link-80 has no overlay capability. In order to use the overlay features of PLINK-II, you need make no modifications to your FORTRAN programs other than to compile your routines separately. PLINK-II will then take care of all overlay assignments in accordance with your commands. No knowledge of memory mapping is necessary; the diagrams in the review were provided as a visual aid. I hope I have helped to clear up your misunderstanding.

Harris Landgarten



COMPUTERS ON TV!

Here is the schedule for the new Computerworld television series.

City	Station	Channel#	Day	Time
New York	WPIX	11	Saturday	9:00 AM
Chicago	WSNS	44	Tuesday	6:30 PM
Los Angeles	KWHY	22	Saturday	12:00 Noon
San Francisco	KTSF	26	Saturday	9:30 AM
Boston	WXNE	25	Wednesday	10:30 PM
Washington,DC	WDCA	20	Saturday	12:00 Mid.
Philadelphia	WTAF	29	Monday	10:30 PM
Dallas/Ft.Worth	KNBN	33	Tuesday	5:30 PM
Detroit	WXON	20	Saturday	10:00 AM
Minneapolis/ St.Paul	KMSP	9	Saturday	TBA*
Houston	KHTV	39	Monday	12:00 Mid.
St. Louis	KDNL	30	Sunday	5:30 PM
Atlanta	WANX	46	Tuesday	11:00 PM

*Check local listings for time.

The Pipeline

by Carl Warren

Expect a 'Soft' Year
Current indicators are pointing to further softening of the economy; but software vendors and designers can plan on enjoying a different type of 'soft' economy.

Already, hardware vendors are realizing that the 'iron' is useless without software. Moreover, most have come to the conclusion that not just any software will do. Manufacturers such as Xerox with the Model 820, for example, are touting its software power rather than the relative unexcitement of the ubiquitous Z80 microprocessor.

What makes systems like the 820 even more exciting is that the software is built around Digital Research's CP/M operating system. This coupled with the growing number of system and applications products for the powerful O/S make the system even more attractive.

The Xerox system isn't isolated either. The normally conservative Hewlett Packard has recently introduced a Z80 based system that employs CP/M, called the HP-125. This decision was made, says an HP spokesman, because HP felt that CP/M and several associated products met the high quality demands that HP customers expect. Furthermore, it opens an application door that was previously closed.

With companies like Xerox and HP already off and running with CP/M compatible systems, it won't be surprising when IBM introduces its compatible system. The IBM product, however, won't be based on the S-100 bus as previously rumored by one East coast wag, nor will it employ Digital Research's brand of O/S.

The system, reportedly named the 5105, employs Intel's 8088 microprocessor and Microsoft's UNIX like DOS. As interesting as this is, in comparison to the other entrants, sources close to the so-called Boca Raton project (named after the IBM facility location in Florida), indicate that the new machine can employ Digital Research's CP/M-86. They

(continued next page)

are quick to point out though that it probably won't make any difference since both the Microsoft O/S and CP/M sport almost identical commands.

Electronic worksheets seem to be popping up all over the place ever since Personal Software convinced the buying public they needed one. The latest bean counters' tool is Sorcim's Supercalc.

The powerful package is CP/M compatible, sports a windowing feature so you can keep your visual place in the matrix at all times, and even supports a digital plotter. Although Richard Frank, the program's designer, borrowed from VisiCalc for inspiration purposes that's where the similarity ends. Already numerous consultants are touting the Sorcim package as the next major software package to come on the scene. Apparently they (the consultants) have reason to make such claims. Xerox for one has latched onto the worksheet as its answer to the well-entrenched VisiCalc. HP, however, has implemented HP-VisiCalc on the 125; but sources speculate that Supercalc will be offered later simply because of its enhanced features, which include dynamic panelling of the hardcopy output.

Sorcim isn't the only one getting on the spread sheet bandwagon. Microsoft has been working on 'Electronic Paper' for several months, and expects to offer it late this year to the general public. Although details are sketchy, the EP product will not only offer expected spread sheet functions such as tabular calculations, but will work in tandem with a word processor, a database manager which is also under development; it will also provide for high-level graphics.

This attention to a total product development is not surprising from the Bellevue-based WA company. This change in product strategy most likely belongs to the consumer division president Vern Raburn. Vern has always believed that a total product offering was worth more than one that performed a single function.

Further enhancing the Electronic Paper product will be numerous application overlays, and support documentation being developed by a subcontractor in the Santa Clara CA area. From what can be learned, users will be able to purchase support packages that will perform a variety of jobs—much like the predesigned support module developed by HP.

Although Sorcim and Microsoft will most likely be the leaders in CP/M compatible spread sheets, others have intentions of following along. For example, Microware, and Technical Systems, makers of 68xx compatible software, are expected to enter the spread sheet marketplace within the next few months.

You probably can expect Personal Software to provide similar support to Tandy's TRS-80 Color Computer, but don't expect to get an answer if you call to find out. They aren't talk'n.

Communication ware is the talk of the industry especially if you plan to set up a local (Ethernet type) or long distance network. For the latter, software supporting a variety of protocols is required, and for the former small systems users would love to have something.

Unfortunately, the problem isn't trivial and requires further definition of the requirements of such a single wire system. Two starting points may be from California. A company called 3Com Corporation has developed an Ethernet transceiver that appears to provide compatibility to any type of system with just a simple connection. The Menlo Park CA based firm may be willing to work with qualified designers, and who knows, maybe a whole new class of communication ware can be developed.

Rather than wait around to see what anyone else was going to do, Corvus Systems Inc. (San Jose, CA), has taken the bull by the horns, and developed its own micro computer network called Omninet.

The network uses what is termed carriersense multiple-access method (CSMA) to enable data entrance onto the transmission medium. The Omninet uses twisted pair rather than coaxial cable and operates at 1Mbit/s, far slower than the Ethernet system, but faster than other communication systems.

This future thinking company has developed the network to work with a variety of systems, including the Apple, and permits up to 64 systems to be connected on a 4,000-ft. serial link.

Currently, prices range from under $500 to about $800 for the intelligent interface called a transporter. This interface is based on a Motorola 6801 microprocessor and a proprietary gate array.

The concept looks great and may open a number of doors for exciting network ware.

But even with the advent of in-house networks, a great deal of work still is needed to support communication over standard telephone lines. With the profusion of protocols, confusion exists among designers about what to use.

San Francisco based consultant Jim Edlin thinks that eventually standardization will arise out of the confusion and a high-speed protocol will result. He further asserts that ASCII transmission will be done away with because of the associated slowness and inefficient use of the transmission media. Others disagree, though, pointing out that straight ASCII communication, without a protocol, is the standard and will most likely sustain small system communication for a long time.

Currently, no one has the answer, nor does it appear that one is quick in coming. What is evident though is that current communications are slow and inefficient; but they are faster than what was available just a few short years ago.

Because of the interest in communication, numerous manufacturers are taking a look at small portable semi-intelligent terminals.

These really small systems, planned to sell for less than $1000, offer direct connect and acoustic coupled capability, and fairly sophisticated editing functions.

Already, Sony, Novation, Trendata, and others have developed products to test the market. What isn't known is whether or not the traveling salesman or executive is willing to throw

one in his attaché in order to do work later in his hotel room, or while riding on a plane.

A more thought provoking question is: "What type of work will the user be doing?" Most of the manufacturers believe that he or she will want to perform order entry, transcribe notes, or write important letters. All of which takes software. This brings up the question(s) of who will write it and to what specifications.

Most system designers, and industry consultants believe that the real inhibitor to the growth of the small portable terminal market will be software. All point out that this is a traditional problem, and usually it is because someone failed to ask the users what they wanted.

ZOSO

New York City - May 29, 1981

Good Day,
It seems like I just returned from Chicago, but I'm headed there again (for the Summer CES - Consumer Electronics Show). This time I have a confirmed reservation for a primo suite in a four-star hotel, so barring extreme insomnia or wanderlust, it's not too likely that I'll make it to Calumet City to visit any of the new friends I made last month (with the possible exception of Debbie the erotic dancer).

New York City - June 14, 1981
I'm back from CES. As was the case last month (NCC), not too much merits reporting. I did notice that there were very few new models of portable stereos (AKA 'disco boxes' or 'ghetto blasters'); in their place was a glut of 'WALKMAN' (TM Sony) type personal listening devices. On the surface, this would suggest a welcome and overdue improvement, but I wonder, and so should you. Public use of [any!] portable music system is a virtually guaranteed indicator of sociopathic tendencies.

Noise polluters circa 1975-1980:
Let's call them DBs (for their Disco Boxes). In between periods of imprisonment, these charming specimens of belligerent truancy just love to provide free (their freedom; certainly not yours) 130 db concerts consisting of distorted Disco or Salsa, an equally nauseous clone thereof (the preferred use of 'nauseous' — you girls from Long Island should look it up if you don't believe me). Where was I? Oh yeah, I'm no fan of technology when it allows entire neighborhoods to be victimized for the price of a few batteries.

The only big winners I perceive are the manufacturers who export these blunt weapons [mostly] from the Orient. On a lesser scale, some local poultry farmers are winners too, because when one of my peaceful evenings is interrupted by a 'Get down, Baby, boogie in my Salsa' serenade beneath my window, the guy responsible will likely sport a free (my freedom not his) six egg 'omelet Tartare' on his inconsiderate little head. It is hoped that my minimal response has some effect, because as the old saying goes, "Today's raw egg is tomorrow's boiling oil."

Noise polluters (circa 1981):
On the other side of an equally unspendable coin we have the nicad powered nincompoops who appear to go about their own business (generally little more than rollerskating or bicycling in traffic), enraptured by whatever they are hearing through their private, foam-padded earphones. Let's call them PEs (for Private Earphones).Trust me, the psychological profile of the PE clowns is almost as obnoxious as that of the DB people, because the PEs generally listen to halfway decent music, but they smugly keep it all to themselves. Few things are more irritating than hearing [just the] crescendos of good music leaking through the earphones of people who think they are making a 'good citizen' statement. Picture if you will one of these PE people wheeling down the street, unaware that a runaway bus is closing fast, because 'La Traviata' is coming through tasteful little earphones ever so much louder. I'm the bystander who won't manage to scream a warning, possibly due to stage fright. I'll work on it . . .

June 18, 1981
Well, I guess it's about time for some investigative reporting; I think I'll dial a few familiar numbers:
5:15 PM EST - 415 887- 8080
(Osborne Computers)
This call provided a pleasant surprise. I was sniffing around with some deliberately pointed inquiries. To my complete delight, a charming and very well informed lady answered all my questions about technical 'specs', pricing, local dealerships and firm delivery dates. She did so politely and to my complete satisfaction. Good Show!

Aside: There are a lot of people who would like nothing better than to see Adam Osborne fall on his face with his new computer ventures. I am not one of them! This gentleman's contributions to the microcomputer industry are immeasurable! I wish to go on record right now as wishing Doctor Osborne the best of luck and continuing success! (Especially since writing acerbic columns was becoming an overcrowded field).

5:45 PM EST - 217 359-2112
(Supersoft)
Whew! I hung up just in time (before the phone rang). I almost forgot that I'd promised never to call this number again (Lifelines March, '81). I was going to ask these guys why they persist in misspelling 'diskette' (as 'discette') in their ads. I also had a couple of questions about their scandalously overpriced (at $100) 'Disk Doctor' package.

One thing which made me curious was this statement in the manual introduction: "DISK DOCTOR typically recovers 87% of all files from disks with directory failures." I'm the kind of skeptic who doubts if any percentage can be reliably plugged into such claims, but at least they kept the number under 99.44% (Ivory Soap, kiddies).

I also wanted to ask why Disk Doctor must offend the user with failed attempts to be amusing. Your problem disk (not disc!) is metaphorically cast as an ailing patient. Should the patient be admitted to the hospital? Do you [as legal guardian] consent to surgery? Do you consent to transplants? (haphazard vivisection) . . . I kid you not!

Even worse, the Disk Doctor will not perform even the most cursory of examinations until 'he' has been 'installed'; (which might suggest returning from the golf course, demanding full payment up front, etc.). A patient whose CP/M system supports system tracks of a different format than the data tracks (and this is not uncommon) will surely die when the Disk Doctor's unwieldy scalpel reaches track 2. Under circumstances more to the Doctor's liking, installation will race along like an antique auto with four flats . . .

We physicians share a conspiracy of silence when it comes to impugning the skills and credentials of our colleagues, but Dr. Zoso has been your friend far too long to stand on tradition. So regarding Disk Doctor (and strictly off the record): His physician's license was issued in Honduras, and I'm beginning to suspect it was forged! What the world needs is a 'Disk Lawyer' to prevent this untrained charlatan from performing further witch doctory, by keeping him in court, answering a lifetime's worth of malpractice suits.

6:25 PM EST - (?)
Paydirt! Paydirt! (Transcription follows): No it doesn't! I have been requested to delete this section for humane reasons, and I have uncharacteristically agreed to do so. Please forgive me for this; I promise it won't happen very often. Eagle-eyed sleuths may already have deduced that when I placed this call, normal business hours were already over in several key time zones, and that's the only clue you're going to get.

July 1, 1981
Bowdlerize (v.t.): To expurgate or modify by removing passages prudishly considered immodest. [After Thoma Bowdler (1754-1825), English editor of an expurgated edition of Shakespeare].

This word has been running through my mind all day long; bear with me and I shall explain.

I just got my copy of the July Lifelines, what an embarrassment! My last column bore little resemblence to what I submitted; (your attention is directed to the 'OOPS' section elsewhere in this issue). The fault lies with the godawful automatic typesetter they've been using here. This bowdlerizing machine (BM for short) apparently has some undocumented text deletion and transposition capabilities, which is how my last article got clobbered. The unit in question is a CRTronic, made by Mergenthaler (of Melville, NY). Make no mistake, this wretched 'Murphy's Engine' will be on its way back to Melville the very second the trial lease expires (around the time you read this). I earnestly advise any of you who

may be investigating typesetting equipment to investigate elsewhere. To restate this valuable advice, you could buy a BM from Mergenthaler for $18000 to $30000, depending on options, or you could march behind a circus elephant until it 'fouls the footpath'; either way your typesetting won't get done very well.

July 2, 1981
To date, I have 'left the door open' as it were, to see if this column might provide a worthwhile clearinghouse for consumer feedback, and I'm not too encouraged by the early results. (Two notable exceptions were provided by Paul Sparks and Leonard Gogol).

Mr. Sparks has volunteered the name of a gentleman who purportedly has the parts and know-how to get discontinued Intertec products up and 'running' after they break down. (I say purportedly because I have cut all ties with that company's product line, discontinued or otherwise, and thus refuse to check this tip out myself): Mr. David Larrison, c/o 9 Associates, 10680 Main St., Fairfax, Va. 22030; (703) 273-2567.

Len Gogol sent me a thick packet containing copies of letters and [his] telephone bills documenting the ongoing nightmare he has experienced, trying to get Omega Research (of Linden, CA) to respond to some [quite legitmate] complaints. I happen to know Len Gogol personnally, and take it from me, you won't find a nicer or fairer guy. I don't know anything about the SHIVA operating system which Omega sells, but I do know that the answers cannot be found in Len's SHIVA manual.

So much for good consumer feedback; what about the rest? Most of it consists of ill-founded gripes. I'm still open to the role of consumer advocacy, but only if the complaints I hear are reasonable.

In virtually every instance, reports of dissatisfaction arise from one or more of the following:

1: A Vendor's dishonesty
2: A Vendor's avarice
3: A Vendor's non-support **or**
4: A Consumer's incompetence
5: A Consumer's dishonesty

Please, if you contact me, make sure that numbers 4 and 5 (above) have contributed nothing whatever to your unhappiness.

ZOSO

Tips & Techniques

Here are two tips that Kelly Smith from CP/M-Net News* (3055 Waco Ave., Simi Valley, CA 93063) has been so kind as to let us reprint. The first tip is from Vol. I, No. 1 (Jan. '81) and tells you how to change logged disks in DDT or SID:

"Have you ever fired-up Digital Research's DDT or SID program debuggers, specifying the program to debug? It loads, then goes '?', because it can't find the file, and it's really on the other diskette (let's say B:) in your system!?! So you have to Control-C out (back to CP/M) and PIP the program to the proper diskette . . . ARGH!

So use DDT or SID to CHANGE the logged-in drive number and get to the file you want . . . Let's assume that we are logged on to the A: drive, and our target file to debug is on B: . . .follow along:

```
A› DDT BUMBFILE. COM«cr»
DDT VERS 2.0
?
-
```

So here we sit (not the least bit amused), contemplating our navels . . . DO THIS!

```
-IBUMBFILE.COM«cr» { set-up temporary FCB with filename.typ }
-S5C«cr»           { Substitute starting at address 5C Hex  }
005C  00  02«cr»   { set drive number 2 (B: disk)  }
005D  42  .«cr»    { quit substituting  }
-R«cr»             { Read BUMBFILE.COM  }
```

As if by magic, the debugger will log on to the B: disk, grab the file, and read it in for your debug session! All you have to remember, is that at address 5C Hex is the start of the temporary file control block, and that:

01 equals A: disk
02 equals B: disk
03 equals C: disk
.
.

So on, and so on for up to 16 disk drives."

Kelly's second tip is from Vol. I, No. 5 (May,'81) and is a seven byte program to discourage TRACE with DDT or SID: (but not for use with CDOS)

"Here's a simple routine to add to any program to discourage anyone from tracing (and revealing) your program's internal operation. In its simplest form:

```
check$ddt$or$sid:   ; check if program is
                    ; being looked at
    lda  6          ; get content of address 6
    ora  a          ; set flags
    jz   banzai     ; use whatever means you think
                    ; fit to clobber this guy
                    ; continue normal execution
                    ; of program
```

This short routine works as follows: Address 6 is the (normally) least significant byte of the address to be called for any operations of the CP/M BDOS (the CALL is at address 5), and as such its value is 06 Hex. Digital Research's DDT and SID debugger programs, when present in the system, always leave a CALLing address on an EVEN page boundary in memory at address 6. This means that address 6 will have a 00 Hex byte for the least significant byte of the address, so this routine simply checks to see if the address is 0! Now, if it is, you can BANZAI the continued operation of the TRACE by. . . hmm, erase the disk directory (?) or erase ALL of memory (?). You get the point. To be really clever with this, save the result of address 6 and let the jerk trace for awhile to lose track of why you did the LDA 6, then set the flags, run awhile longer, then (after he/she has forgotten that also) check the result of the flags.

P.S. This will not work with FAST running in conjunction with your program, so watch out! Also, my preferred method to BANZAI the person trying to "tear apart" my software is to scramble the program, then write it back out to disk. Remember, that BDOS CALLs cannot be traced anyway. So about the time that the "tracer" realizes that file open/write/close occurs, it's just too late!"

*CP/M-Net News is printed monthly (at worst quarterly) at a yearly subscription rate of $18; payment must be in advance, by check or money order.

Large Memory Management Comes to the Z80

by Bernard P. Wess, Jr.-VP, Product Development, CSSN, Inc.

Boston, MA. CSSN, Inc. has introduced a desk-top microcomputer capable of addressing 1 megabyte of RAM with a Z80 processor. Configured for commercial data processing and scientific computing, the Stretch 1000 is designed especially for use with database management systems.

The Stretch 1000 system provides an answer to concern voiced over many Z80 microcomputers. The Stretch system provides the system developer or OEM with a fully integrated computer system, designed with emphasis on large program space, mass storage capacity and flexibility, performance, serviceability, database management, mathematical processing.

We [at CSSN] recognize and have solved the problems associated with program size on a Z80 based computer. The Z80 is the industry dominant leader in processor technology and we are committed to maintaining and enhancing its value, but the 64K addressability is inadequate for many applications, and for our microcomputer operating system, which is not based on "floppy" disk technology.

The Stretch system was conceived as an operating system and hardware combination that would support single-user database management. So we have borrowed well-proven memory "mapping" technology from minicomputer designers. At the same time, this technology was implemented so that each task in memory can be protected. The operating system, its disk directories and buffers, as well as other tasks (for example, a database management system [DBMS]) can be protected from destruction by another task.

Figure 1 shows a typical application composed of a main program, two overlay areas, and a query language task. The operating system resides in its own 16K segment outside the address space of the application. The DBMS is given a 64K segment for itself and its buffer. The DBMS is aware only of its own 64K address space; thus it can be given maximum buffer allocation, increasing performance substantially.

Applications generally require extended addressing, not for data structures or arrays, but for subroutines and procedures that can be organized as "overlays". The Stretch operating system manages real memory address space as a series of overlays. Overlays are defined by the application programmer and the Stretch overlay manager (termed a "linking loader") will allocate overlays within as much as one megabyte of real address space. The example shows a 112K application area and the DBMS is permanently resident. Adding the DBMS, the actual application and the DBMS are utilizing 208K of real memory on the Stretch Z80.

Any number of overlay areas are possible. The example shows two, but the actual number is determined by the nature and size of the application.

The Stretch overlay manager is fully integrated into the operating system and is aware of the current overlay structure and the available real memory above the operating system. An application programmer who understands the subroutine/procedure hierarchy can quickly define an overlay structure for the Stretch operating system. Figure 2 depicts the relationship between real memory pages and logical memory pages as seen by the Stretch operating system.

The Stretch system fully integrates the operating system and the system hardware to provide a complete development and production computer. Mass storage includes Winchester disk drives of 10 to 169 Mbytes. All disk storage drives are fully integrated with a 13.4 Mbyte tape cartridge for file-by-file archiving of data. The "BackUp" system software conforms to archiving systems found on many large mainframe operating systems. As with all CSSN software products, the BackUp system software implements the finest features of mainframe operating systems and ignores overlay complex conventions or procedures found on these systems.

The Stretch system is a high performance product. Based on the BASF disk drive, Stretch includes fundamental performance enhancements in direct access I/O while still maintaining CP/M compatibility. The Stretch operating system includes a CSSN re-engineered direct access subsystem conforming to CP/M 2.2 direct access but 10-40 times faster depending on which system Stretch is compared with. By conforming to CP/M system calls, the Stretch system assures the OEM a source of compatible systems software optimized for high speed and large capacity mass storage.

Capacity and performance are of little concern if an OEM or his customer is incapable of providing a fast and efficient service of installed computers, particularly if a significant installed computer base exists. Our systems, particularly the Stretch system, are based on the IEEE S-100 (696) specifications. OEM, third-party, or customer personnel can rapidly diagnose a failed component and replace it in the customer's office. The Stretch system is composed of multiple circuit boards and components which are easily accessible, thus easily removed. Our Stretch system is not a single board computer and encourages serviceability by conforming to the S-100 bus.

Extensive microdiagnostics (the System Test Package - STP 1000) are resident within the system and available at a touch of the system reset button. On power initiation, STP executes diagnostic sequences for the processor, all memory, I/O ports, timers, tapes, and disk controllers and the system disk drives. Any failures are reported and isolated and multiple tests are available to the customer if unusual problems occur. Microdiagnostics can be executed remotely by customer service personnel or a diagnostic computer.

For high risk applications or geographic areas experiencing poor

power reliability or quality, an inexpensive Uninterruptable Power Supply (UPS) is available.

Providing support for mathematical/scientific data processing has been a difficult problem for the Z80 micro computer market. We have addressed these concerns of engineering and scientific systems developers by providing the Stretch system with an integrated math processor as an option.

Although each array may be only 64K bytes in length, a sophisticated programmer could implement very large arrays or matrices assigned as FORTRAN COMMON BLOCKS or PL/1 STATIC variables in overlay segments. Very large scientific or engineering applications are then possible while still maintaining CP/M and Z80 compatibility.

Numeric processing will be provided by a subroutine library in C, PL/1, and FORTRAN integrated with the AMD 9512 math processor. Depending on the algorithm the processor provides performance increases as much as 500% in applications requiring numeric computation.

The Stretch 1000 is available in the standard System 1000 desk top computer. The Stretch system comes with the Stretch processor, 24 Mbytes of disk storage, a 13.4 Mbyte tape, and 128 Kbytes of memory. The system is expandable to 1 Mbyte of memory.

System software includes the operating system, the overlay manager, and numerous system utilities. The AMD 9512 processor and the mathematical library are optional.

PERFORMANCE EXAMPLE

A cross reference generator for programs, which generates cross refernece by module and keyword was used to test the performance of Stretch 1000 and MDBS version 1.0. The following example indicates the speed improvements due to the use of ALLOCated direct access files (replacment of CP/M 2.2 direct access) and the additional performance gained by using the Stretch processor.

For a thirty file database with 899 keywords, the following statistics were gathered.

Performance Example

	S-1000 CP/M 2.2		STRETCH 1000 ALLOCated Direct Access	
Buffers	15	30	15	30
Reads	3864	1314	3764	1109
Writes	405	151	396	159
Elapsed Time	70 min.	25 min.	7 min. 15 sec.	3 min. 50 sec.
Performance Factor	1:1	2.8:1	9.6:1	18.4:1

Stretch 1000 Configuration

- Basic System
  - S-100 IEEE-696 8 slot bus
  - 128K RAM 4Mhz
  - 2 Parallel Ports
  - 2 Serial Ports (I10-76,800 bps) RS-232C
  - 13.4 Mbyte tape drive with ANSI/ECMA
  - 1/4" data cartridge (3M DC300XL) 6250 bpi
  - Tape controller supporting up to 4 drives
  - 24 Mbyte 24 m Sec BASF disk drive
  - Disk controller capable of supporting 4 drives (SMD Interface
- Characteristics
  - 9" high X 19" wide X 29" deep
  - Weight: 110 lbs.
  - Electrical Specifications: 500 Watt, 110/220 volts, 50/60 Hz AC ± 10%
  - Humidity: 20-80% non-condensing
  - Ground: National Electrical Code required
- Options
  - 10 to 169 Mbyte drives
  - AMD 9512 Math Processor
  - 128K to 512K RAM memory board
  - Up to 1 Mbyte RAM
  - Communication Processor card
  - CRT terminals
  - Printers
  - 9 track 1600 BPI IBM compatible tape drives
- Software
  - Math Library available
  - MDBS configured for Stretch 1000
  - Overlay manager
  - PL/1
  - COBOL
  - Tape drive support for RTL Audit Trails and recovery

(continued next page)

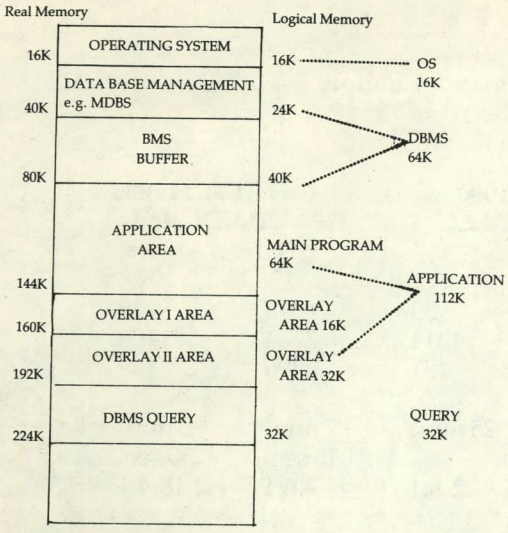
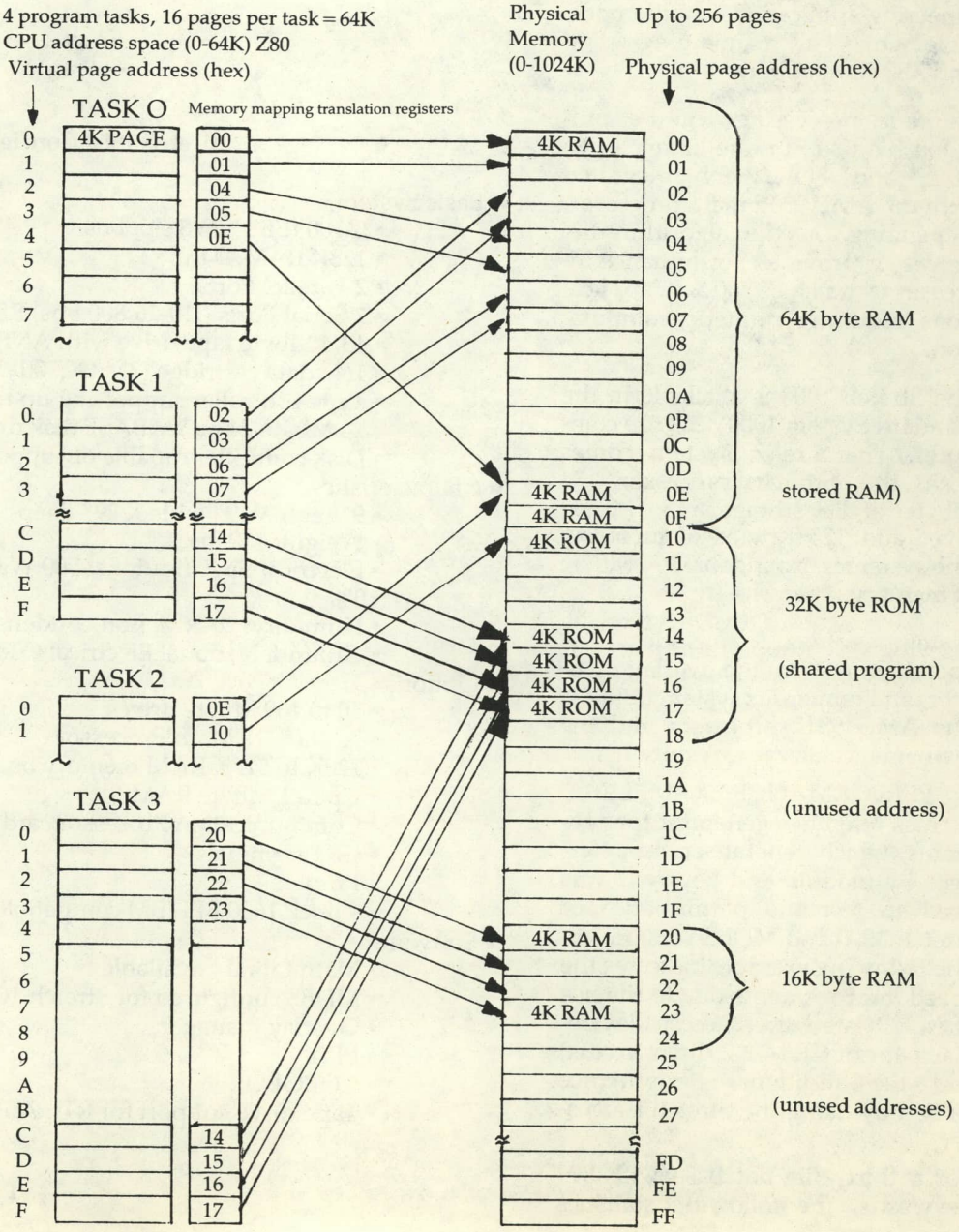


FIGURE 1: A typical application using the Stretch 1000 processor

FIGURE 2:

CSSN MEMORY MAPPING ILLUSTRATION



CP/M Bit Map File Allocation

by Kelly Smith

What It's for

For each diskette "logged-on" to the CP/M Basic Disk Operating System (BDOS), physical diskette space is dynamically allocated to that diskette (for later write operations) and maintained in memory by Bit Mapped Allocation. When changing diskettes (and to avoid the "R/O" error message) you must enter Control-C to erase the previous diskette's Bit Map, and cause the BDOS to read the file directory to establish the new Bit Map. On subsequent write operations to the diskette, the Bit Map is modified, and the diskette File Control Block (FCB) for the (now closed file) is updated in the File Directory.

What It Is

The Bit Map is actually a tight encoding of available (or not) sectors on the diskette. The Bit Map is an array of single bits which correspond to each block of eight sectors allocated for usage on the diskette. A blank (formatted) diskette's Bit Map Allocation array then looks like this:

("Standard" Single Density IBM Format Diskette)

GROUP ALLOCATION MAP DRIVE - B

```
11000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000
```

240 GROUPS REMAINING ON DISK OUT OF 243

Notice that two bits are 1's . . . this predefines allocation space for the CP/M Directory for the diskette (i.e., 16 sectors or 2 groups), and insures that the directory is not overwritten when creating new files. All remaining 0's are free groups, ready to become allocated in the directory. When the BDOS receives a request to create a file, it first searches the Bit Map until it finds a bit containing a zero, and the number (this will be explained shortly) of this bit is the number of the first free group to be allocated for the file. The BDOS sets the bit map to a one and places a one byte hexadecimal group number into the FCB of the directory, created for the new file. As subsequent write operations occur for the file, the BDOS examines the last group number in the FCB (and also the Next Record count) and from the numbers automatically computes the next physical track and sector number where the diskette write is to occur. Keeping in mind that eight sectors equal one group, when all eight sectors of a group have been written, the BDOS searches the Bit Map again for the next bit containing a zero. When a free allocation is found, its group number is added to the FCB (not necessarily a sequential number) and the corresponding bit is set to a one. Also note that the minimum file size that a file will be, is one kilobyte . . . a file which has seven or fewer sectors will be shown (by STAT Filename. Typ«cr») as utilizing one group (1k); in CP/M 1.4 a file which has eight sectors will be shown as utilizing two groups, even though the second group is empty.

Here is an example of what one sector of the directory looks like, showing four FCB's for three files:

Filename:	.Typ	EXRC	Group			
MACRO	.LIB	00 80	45464748	494A4B4C	4D4E4F50	51525354
MACRO	.LIB	01 08	55000000	00000000	00000000	0000000(
PLINK	.COM	00 0B	56570000	00000000	00000000	0000000(
DISKTEST	.COM	00 09	58590000	00000000	00000000	0000000(

Specifically, note that the file MACRO.LIB has two entries . . . and also the Record Count (RC) for the first entry is set to 80 hexadecimal. As new records are written to the diskette, the RC is updated by the BDOS. When a transition occurs from 7F to 80, the BDOS adds a new FCB into the directory, and also creates a new extension (EX) to the file. Bit Map allocation then proceeds from the next available group.

And Why

As I mentioned, the group allocations may not be sequential (even though my example shows sequential groups). As files are deleted from the diskette, they leave "holes" in the Bit Map to make space available for any new files to be created. The major advantage here, is that the disk is never required to be "packed down" (iCOM FDOS users or UCSD Pascal users know what I mean!). You users of an operating system that utilizes Sequential or Linked Allocation methods, my heart goes out to you . . . nothing quite so "gut wrenching" as a disk error in the middle of a pack operation! The other advantage to Bit Map Allocation, is true random access to sectors (no "kludge" ISAM, as in MITS DOS or BASIC and no "rewind file pointers" as in PASCAL), and dynamic allocation of file size. Another source of frustration regarding Linked Allocations occurs when an isolated sector (which "forward references" the next sector as a disk error in the middle of a pack operation! The other advantage to Bit Map Allocation, is true random access to sectors (no "kludge" ISAM, as in MITS DOS or BASIC and no "rewind file pointers" as in PASCAL), and dynamic allocation of file size. Another source of frustration regarding Linked Allocations is when an isolated sector (which "forward references" the next sector in the file) is "bombed" . . . there is no way to recover the ENTIRE file intact . . . just up to the point of the offending sector!

Now for some detail (as promised): To determine the group allocation from the Bit Map, we must first "split" the group in half . . . then it all begins to make (I hope) good sense. To find the (hexadecimal) group number of

(continued next page)

an individual bit in the Bit Map, take the first digit from the even (or odd) half row and the digit from the column in the same half. As shown in the example below then, the next free group allocation is 02 and the last free group is F1:

GROUP ALLOCATION MAP DRIVE-B

	0123456789ABCDEF	0123456789ABCDEF	
0	1100000000000000	0000000000000000	1
2	0000000000000000	0000000000000000	3
4	0000000000000000	0000000000000000	5
6	0000000000000000	0000000000000000	7
8	0000000000000000	0000000000000000	9
A	0000000000000000	0000000000000000	B
C	0000000000000000	0000000000000000	D
E	0000000000000000	00	F

240 GROUPS REMAINING ON DISK OUT OF 243

The PHYSICAL relationship of sectors to groups is such that Group 02 starts at Track 2/Sector 20, Group 03 starts at Track 02/Sector (YES! Sector 13, because of the diskette "skew factor" . . . hmmm, a good topic for yet another mundane article on my part . . . OK, OK, . . . BORING!), and so on.

In conclusion, Digital Research's CP/M BDOS makes the most efficient use of available space on the diskette. Other operating systems (let's cuss MITS DOS again!) often require that the file size be specified by the user . . . overcaution (and much guessing) results in large amounts of unused diskette space that is NOT AVAILABLE to other files. This space can only be recovered by copying the data to a new diskette with the proper file size specified, and WHAT'S EVEN WORSE is that this same procedure must be followed to EXPAND a file that has already utilized the space originally allocated to it! So anyway . . . NICE JOB Dr. Kildall!

This article is reprinted from Volume I, No. 3 (March) of The CP/M-Net News. See page 7, Tips & Techniques, for details.

A Patch For ZDT

ZDT Version 1.41 permits the user to load the primary Z80 registers through depositing values in (undocumented) memory locations. This is particularly useful for making a permanent modification to ZDT so that it comes up with the Program Counter set at 0100H, rather than at 0. Below are the memory locations and the registers filled from these locations:

Register	Location
HL	0f96H
DE	0f98H
BC	0f9aH
A	0f9dH
PSW	0f9cH (right nibble sets HPNC flags, leftmost two bits set SZ flags)
SP	0f9eH
PC	0fA0H

After making the modifications, they can be be saved by returning to CP/M and typing "SAVE 24 ZDT.COM".

A Bug In the Z80 Chip –A Lifelines Exclusive

New York, N.Y. News has reached Lifelines that Robert Burns of Balcones Computer Corporation in Austin, TX has discovered a bug in the Z80 chip.

This defect is known to occur during the execution of the INI and INIR instructions. It is probable that the bug also affects other similar instructions. The INI instruction inputs a byte of data from the port number contained in register C, places the input data at the location addressed by the contents of the HL pair, increments the HL register pair and decrements the B register. The INIR instruction repeats this cycle until register B is zero. The documentation of these instructions clearly states that the carry flag is not affected. The carry flag, however, is affected in many cases, the result apparently depending upon the port input from. The code segment:

```
LD HL,200H      ; DESTINATION STARTS AT 200H
LD B,OFFH       ; TRANSFER FF BYTES
LD C,7DH        ; USE PORT 7D
INIR
RET
```

will cause the carry flag to be set regardless of its condition prior to the execution of the INIR instruction. The Mostek and Zilog versions of the Z80 are both affected. Mr. Burns discovered this bug while rewriting some disk driver routines which used carry set as a disk I/O error indicator. After substituting the INIR instruction for less efficient 8080 code, all calls to disk I/O primitives returned with carry set, indicating an I/O error.

Since many routines use the carry flag as a status indicator, one should be careful to avoid time-consuming surprises when using the indirect I/O instructions.

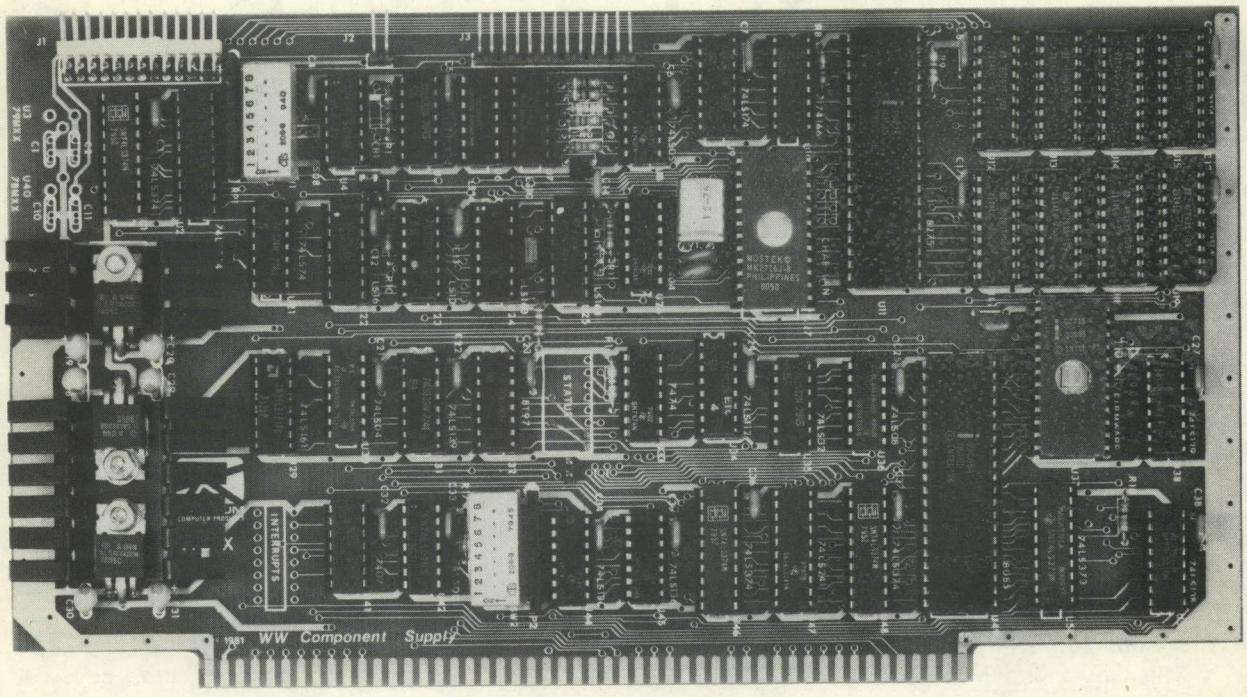
Harris Landgarten

RENEW

Subscribers who began getting Lifelines in September (Volume I, No. 4) must renew very soon in order not to miss next month's issue. You'll be receiving a notice in the mail–so heed it! When renewing use the form you have received in the mail, or enclose a Lifelines mailing label. This will help us process your order more quickly.

When informing us of a change of address or a correction, please send along an old Lifelines mailing label. We must have your precise old address in order to expedite your requests.

INTELLIGENT VIDEO I/O FOR S-100 BUS



VIO-X

The VIO-X Video I/O Interface for the S-100 bus provides features equal to most intelligent terminals both efficiently and economically. It allows the use of standard keyboards and CRT monitors in conjunction with existing hardware and software. It will operate with no additional overhead in S-100 systems regardless of processor or system speed.

Through the use of the Intel 8275 CRT controller with an onboard 8085 processor and 4k memory, the VIO-X interface operates independently of the host system and communicates via two ports, thus eliminating the need for host memory space. The screen display rate is effectively 80,000 baud.

The VIO-X1 provides an 80 character by 25 line format (24 lines plus status line) using a 5 × 7 character set in a 7 × 10 dot matrix to display the full upper and lower case ASCII alphanumeric 96 printable character set (including true descenders) with 32 special characters for escape and control characters. An optional 2732 character generator is available which allows an alternate 7 × 10 contiguous graphics character set.

The VIO-X2 also offers an 80 character by 25 line format but uses a 7 × 7 character set in a 9 × 10 dot matrix allowing high-resolution characters to be used. This model also includes expanded firmware for block mode editing and light pen location. Contiguous graphics characters are not supported.

Both models support a full set of control characters and escape sequences, including controls for video attributes, cursor location and positioning, cursor toggle, and scroll speed. An onboard Real Time Clock (RTC) is displayed in the status line and may be read or set from the host system. A checksum test is performed on power-up on the firmware EPROM.

Video attributes provided by the 8275 in the VIO-X include:

- FLASH CHARACTER
- INVERSE CHARACTER
- UNDERLINE CHARACTER or
- ALT. CHARACTER SET
- DIM CHARACTER

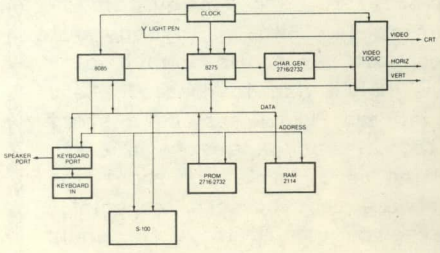
The above functions may be toggled together or separately.

The board may be addressed at any port pair in the IEEE 696 (S-100) host system. Status and data ports may be swapped if necessary. Inputs are provided for parallel keyboard and for light pen as well as an output for audio signalling. The interrupt structure is completely compatible with Digital Research's MP/M ®

Additional features include:

- HIGH SPEED OPERATION
- PORT MAPPED IEEE S-100 INTERFACE
- FORWARD/REVERSE SCROLL or
- PROTECTED SCREEN FIELDS
- CONVERSATIONAL or BLOCK MODE (opt)
- INTERRUPT OPERATION
- CUSTOM CHARACTER SET
- CONTROL CHARACTERS
- ESCAPE CHARACTER COMMANDS
- INTELLIGENT TERMINAL EMULATION
- TWO PAGE SCREEN MEMORY

VIO-X1 - 80 × 25 5 × 7 A & T **$295.00**
Conversational Mode
VIO-X2 - 80 × 25 7 × 7 A & T **$345.00**
Conversational & Block Modes



VIO-X S-100 I/O INTERFACE

FULCRUM™ COMPUTER PRODUCTS

Distributed by:
WW COMPONENT SUPPLY INC. 1771 JUNCTION AVENUE • SAN JOSE, CA 95112 • (408) 295-7171

CPMUG News

by Jim Mills

Well, good ole Jim has been sluffin' off the last few issues -- meaning that I haven't gotten ambitious enough to write an article! Actually Ward says most everything that needs to be said when he writes an article, and the catalogs and abstracts say the rest, so...

This month we present volume 52, COPYFAST 3.5 and BATCH/VARBATCH. Copyfast is Chuck Weingart's fast copy program. Make sense? It is controller dependant, but only in that it uses different interleave speeds. It does do BIOS calls for the I/O, so you don't have to be a "hacker" to use it (If you don't know what a "hacker" is, you aren't). See the abstracts as well as the assembly code source file comments.

BATCH and VARBATCH are submissions to The Users Group by Dan Ross of Succinct Systems in Santa Cruz, California. They are, simply stated, super SUBMIT utilities. The main difference between BATCH and VARBATCH is that VARBATCH does everything BATCH does, plus more, so I don't know why anyone would want to use plain old BATCH, but mine is not to reason why, etc. The only improvements that I would make to BATCH are (1) to make it self-relocatable and (2) to make the generation of the .BAT (as opposed to .SUB) file more readable. If anyone undertakes such a project, please submit it! We can use all the submissions we can get.

Speaking of submissions, we have SOME stuff still in the Mill, but we are always looking for more material to be reviewed for inclusion in CPMUG. Anybody out there got any public domain software they'd like to donate? Send it in, and PLEASE include a short note, either paper or on disk, telling us a little bit about the software and yourself, especially where you got it from. We would appreciate it, makes our job much easier.

Well, that's all folks!

The CP/M Users Group Volume 52 Catalogue and Abstracts

DESCRIPTION: COPYFAST V. 3.5 by Chuck Weingart
BATCH/VARBATCH by Daniel Ross

NUMBER	SIZE	NAME	COMMENTS
		-CATALOG.052	CONTENTS OF VOL. 52.
		ABSTRACT.052	Abstracts of files on this disk.
		CRCK.COM	CRC program (see abstracts).
52.1	12K	BATCH.1	SIL80 source for BATCH.
52.2	10K	BATCH.3	SIL80 source for BATCH.
52.3	2K	BATCH.COM	Object file of above.
52.4	1K	BATCH.CTL	Control table for batch.
52.5	5K	BATCH.HEX	Hex file of above code.
52.6	35K	BATCH.RM	Batch Reference Manual (.DOC).
52.7	21K	BATCH.SYM	Batch Symbol Table.
52.8	2K	CPYFST35.ABS	Chuck's comments on Copyfast.
52.9	26K	CPYFST35.ASM	Source code for COPYFAST.
52.10	28K	CPYFST35.DOC	Documentation for COPYFAST.
52.11	2K	CPYFST3F.COM	Object of above, fast disks.
52.12	2K	CPYFST3M.COM	Object of above, medium disks.
52.13	2K	CPYFST3S.COM	Object of above, slow disks.
52.14	1K	CRCKLIST.CRC	CRC list of programs on this disk.
52.15	2K	EXAMPLE.BAT	Sample BATCH file.
52.16	23K	VARBATCH.1	SIL80 source for VARBATCH (BATCH with variables).
52.17	10K	VARBATCH.3	SIL80 source for VARBATCH.
52.18	3K	VARBATCH.COM	Object of above.
52.19	1K	VARBATCH.CTL	Control table of above.
52.20	7K	VARBATCH.HEX	Hex file of above.
52.21	29K	VARBATCH.SYM	Symbol table of above.

VARBATCH is documented in BATCH.RM (Reference Manual).
COPYFAST is documented in the .ASM files.

BATCH and VARBATCH are some very nice programs by Dan Ross of Succinct Systems in Santa Cruz, California, which is the place from which you can order SIL80Z80 symbolic "assembler" if you want ($150 as of June 1981). Both Batch and VARBATCH are public domain and donated to the UG by Mr. Ross. Essentially, BATCH and VARBATCH are super SUBMIT programs, although they take a little getting used to. Read the documentation (BATCH.RM) carefully to familiarize yourself with the usage procedures.

WARNING: some double-density systems warm-boot ALL of CP/M from disk and wipe out some BIOS changes made by BATCH, so be careful of this. Mr Ross has included a "troubleshooting" guide at the end of the BATCH.RM file. The programs work fine, as-is, on my single-density Tarbell controller system using the BVIOS/BVBOOT (CP/M UG Vol. 38).

BATCH is the submit utility without variables. VARBATCH includes variables 0 thru 9 for internal use, as well as some other goodies. See the documentation for details, as well as usage instructions.

Abstract by Jim Mills

CPYFST35.ASM	
CPYFST35.DOC	
CPYFST3F.COM	FOR FAST CONTROLLERS (56 SEC.)
CPYFST3M.COM	FOR MEDIUM CONTROLLERS (67 SEC.)
CPYFST3S.COM	FOR SLOW CONTROLLERS (78 SEC.)

CPYFST35 is version 3.5 of COPYFAST (CPMUG VOL. 47), a con- troller independent disk copy program for CP/M. Copyfast is an adaptation of the Tarbell disk copy program, but does not require Tarbell hardware. Version 3.5 contains several small bug fixes, faster copy speeds, and three new features: it is now possible to CTRL-C out of the program at any time, the range of the copy can be changed by a parameter given on the command line when the program is started, and the program can now determine the size of the CP/M it is running in and adjust accordingly. The .DOC file contains all of the installation instructions and a great deal more about the program and op- tions than Copyfast did. The source can be assembled with ASM or MAC, and is extensively commented. The front of the source has all of the assembly options, and it should be no problem for anyone to modify these options and tailor them to the particular needs of the user. As supplied, the program copies all of tracks 2-76 of a standard 8 inch disk as fast as a controller can go. Assembly options are present to accommodate slower controllers or single disk CP/M systems. The three .COM files are assembled versions for three different read and write interleave combinations, try all three to see how fast your controller is.

Review by Chuck Weingart

Ordering from CPMUG

A number of Lifelines readers have asked that we publish information on how to order from The CP/M Users Group. All orders must be sent through the mail, as CPMUG has no phone. So please don't call. Orders must be prepaid, by check, money order, MasterCard, or VISA. All checks must be in U.S. dollars, drawn on a U.S. bank. The CPMUG catalogue is available for $6.00 (U.S., Canada, Mexico) or $11.00 (all other countries). Further ordering information, regarding disk formats, disk prices, etc. is available in the catalog.

The Software Evaluation Group: Ashton-Tate's dBASE II

by Steve Patchen and Ed Paulette

Name of database package:
dBASE II
Authorship of package:
Ashton-Tate
Address:
3600 Wilshire Blvd
Suite 1510
Los Angeles, Calif. 90010

dBASE II is a fully relational database management system. Of the systems we have reviewed so far, dBASE II compares very closely to Condor Series 20/DBMS Level II. Series 20 and dBASE II differ in some important respects. First, Condor can be said to be a "set-oriented" system while dBASE II is more "record-oriented". Secondly, Condor depends on the facilities of the CP/M SUBMIT command for the development of batch procedures while dBASE II includes an integral programming language. These two major differences have significant consequences in regard to the appearance of the system to the user - in spite of the fact that both systems use the same general data model and that either system could be used in many situations. [The term set-oriented is here used to connote that most commands in Series 20 result in a manipulation of an entire dataset. Record-oriented refers to dBASE II's explicit handing of one or more specific records within a dataset (although in most cases dBASE II commands allow for treatment of all records in the same fashion).] There are other, more specific differences between dBASE II and Series 20, but those will be discussed in a separate section. For more details on relational database systems see the article on Condor in Vol. I, No. 11, pp. 6-2, the introduction to database systems in Vol. I, no. 6, p. 12 and the references at the end of this article.

All of the problems described in our article in Vol. I, No. 7, pp. 2-5, could be implemented with this system. This article will use the general ledger and checkbook problem described in the Condor article to make a comparison of the two relational systems. Overall, dBASE II is an excellent system. All troubles encountered while working with the system were minor and different strategies were available to avoid them. The greatest convenience of dBASE II is the ability it gives the user to implement a complex system a small piece at a time.

The built-in programming language is used to perform all processing steps, data manipulation and listing functions. It is an interpreter and resembles BASIC and some other interpreters such as APL in that commands can be executed immediately or a preset sequence of commands can be prepared in a disk file for later and repetitive execution. (Currently the command files must be prepared outside dBASE II, a minor inconvenience.) Since command files can invoke one another, can pass parameters back and forth (using global "memory" variables) and can be invoked interactively, quite powerful and complex applications are possible for sophisticated users. The primary drawback of the system is that the user must learn this programming language in order to use the system. Simple things can usually be done with simple statements, but the form of the statement is often procedural in nature and may prove a barrier to users with no previous experience in programming.

SETUP

The terminal installation worked well with all the terminals we tried. Even though the system was designed for an 80*24 screen there are only small problems on a SOL20. Editing a field which wraps around the end of line is messy on a SOL20. Data definition is initiated through the CREATE command. A MODIFY command is used to make changes to an existing dataset definition. There are only three basic data types; characters, numerical and logical. A length specification is required for characters and numbers. A decimal place is optional for numbers. The logical type can only have one of the following values; T,F,Y or N. Data can be immediately entered into the created data set or left empty for later use. Data is stored in ASCII form. Thus a 10 digit number requires 10 characters,a logical field 1 character, and a date from 6 to 8 or more characters, depending on how it is represented.

There is an APPEND command to add new records to a file and an EDIT command to make changes to existing records. An INSERT command allows the operator to place a single record at the current position in the file. All three commands can use either a teletype or a full screen entry mode. The programming language which is part of the system allows the user to create a custom screen format to be used for data entry. This is useful for entering data into related datasets and when entries have to be verified immediately under program control.

Data can be output to the terminal or printer using the LIST or DISPLAY commands. The two commands provide an unformatted listing of the contents of a specified set of records in the currently selected data file. A somewhat more readable format can be produced using the more powerful REPORT command which provides a tabular listing of the data, can perform control breaks and can compute totals and subtotals. (A given report format need only be specified once.)

DOCUMENTATION

The manual comes in two parts. The first part is useful for the first time user. It was written by a system user (rather than the programmer), and takes a casual, narrative approach in discussing each feature of the system and giving examples of their use. The second part is more concise (even terse) and is intended as a reference manual. Unfortunately there is no index and the details of some commands are hard to find. This second, reference section should also have more examples of the various syntax possibilities.

DATA ENTRY, UPDATE AND RETRIEVAL

A full screen editing mode is available for data entry and direct updating of file records using the commands discussed above. In addition, the programming language has many features which allow the programmer to create his own data entry routines. Data can be entered into a dataset or a dataset can be updated either from the terminal or from another dataset. The REPLACE and UPDATE commands can be used to make changes in either individual records or throughout a dataset. The REPLACE command is used to replace the contents of specified data fields of the current record. The UPDATE command uses one datafile to update another one by adding to or replacing specified fields. The CHANGE command allows editing of specified fields from selected records.

There are two commands to retrieve individual records, FIND and LOCATE. The FIND command requires that an index file exist for the file being examined. The LOCATE command does a sequential search of the file on a specified field expression. Neither of these commands allows specification of a subset of records to be used in succeeding commands and it is not possible to create an index which only indexes a subset of the records in the dataset. The user must either explicitly copy over the desired subset or re-specify the selection criteria on each command which is to reference it. ("Memory" variables should make this quite simple.)

GENERATING REPORTS

The included REPORT utility is limited to reports of single rows per record in columns, but the programming language has enough printer control and algebraic manipulation of fields and variables to make report writing easy. A typical report requires about one page of program lines.

PROCEDURAL LANGUAGE

The programming language is block structured using WHILE loops and IF - ELSE structures. (There is no GOTO command and hence no need for statement labels.) Statements in the language are identical in form and effect to commands which might be entered interactively. Since command files can invoke other command files to a nesting level of 10 or more, quite modular programs can be developed. Performance could suffer in loops in which command files are invoked since CP/M will have to open and close the file each time. If the invoked command file was not large it could of course be included in the calling program. This feature also means that programs are not limited by available memory size. Short routines can be used locally by loading them into memory variables and using the built-in macro substitution facility to invoke them. (This is also a method for carrying subset specification from one command to another.) Parameters are passed to one command file invoked by another command file via memory variables which function as global variables.

In both direct and deferred execution modes the usual set of arithmetic functions is available as well as boolean, comparison and string operators. Especially important is that the usual order of precedence in execution is maintained and the parentheses characters change the order of execution in the usual manner. (This is not true in the Series 20's COMPUTE statement.) Functions are provided to convert strings to numerics, extract substrings, search a string for the occurrence of another string, convert a numeric value to a string and to test for end of file, deleted records, record numbers and such.

An important feature of the language/system is the manner in which variables are handled. Any given datafile may contain (only) up to 32 fields and two datafiles (the primary and secondary) may be active at any one time. The values of any field of the current record of either datafile can be referenced in a command statement. In addition to the (up to 64) field data values, one can define as many as 64 additional global memory variables, whose values can total as many as 1524 characters. Variables can be deleted and replaced by other variables or redefined as often as necessary, but it is not possible to selectively save and restore their values from files.

This is a significant limitation in the handling of memory variables. If the values of memory variable are SAVEd, the in-memory copies are deleted. If values from a previous SAVE are RESTOREd, any already existing values are also deleted. It is difficult to understand this design. Aside from this problem, however, the memory variable facility is a very powerful and flexible adjunct to the command language.

A critical feature missing in the language is access to the data definitions and data type information associated with the data files. The structure of a data file can be displayed and passed to another newly created file. This is convenient for creating temporary batch files or backup files. It is inconvenient for maintaining a data dictionary. Since the definition of data is not independent of the files which use the definitions, deleting a file could remove the only definition of the dataset from the database.

(continued next page)

COMPARISON TO CONDOR

dBASE II and CONDOR are both quite powerful systems. In one way or another one could accomplish almost anything with one which could be done with the other. Given the generality of the model on which they are based, this means that most data manipulation tasks could be accomplished by either. Each system will appeal to quite different audiences, we think. Condor's set orientation is likely to be more appealing to the new user or the casual (occasional) user or the user who hasn't the time to learn (yet another) computer language. Such a user may find dBASE II's array of language statements and facilities somewhat overwhelming and much of the power of the system will be difficult to harness. The set-oriented concepts will not necessarily be any easier to learn for someone who has had little programming experience. Such a user may also be less demanding of the system with regard to such items as immediate checking of customer numbers, or salesman codes during data entry. Conversely, Condor's high level language makes it very difficult to handle tricky details. Even novice programmers may chafe quickly at some of the more indirect techniques necessary to accomplish what can be done in a couple of dBASE II statements. dBASE II's language isn't hard to learn as languages go, but it is fully a programming language, so expect to spend some time to master it.

Series 20 is conceptually more concise. The commands used to manipulate data are also (where applicable) used to manipulate data definitions. One can PRINT, LIST, SELECT or WRITE records from dataset definition files as well as from the datafiles themselves. Many functions are supplied automatically by Series 20 that must be programmed by the user using dBASE II. A couple of important examples are the screen definition function and the tabular LISTing function (REPORT in dBASE II) provided by Series 20 which are at least more tedious in dBASE II. Conversely, dBASE II is more application orientated. The command set to control the user interface is more complete. Thus an application can be developed to isolate a non technical user from the details of system operation and data flow. A dBASE II program can be written to print on pre-printed forms. Series 20 cannot do this at all without converting to an external data format and using external reporting facilties. dBASE II allows a wide range of menu systems to be developed to drive application programs.

We have yet to compare the performance of the two systems in processing files containing numbers of records in the thousands. In a future update, we hope to provide some comparative figures. It is feared that dBASE II will process slowly when it is executing several commands on each record (as would be typical when a command file is in use), performance may become unacceptable if a command file invokes another command file inside a record-by-record loop because each invocation will require CP/M to open and close a file. These factors may give Series 20 an advantage on larger datasets in some applications. We have provided a table which gives approximate translations of Series 20 commands to dBASE II commands or command sequences. A similar table was started to translate from dBASE II to Series 20 but because of the number of dBASE II commmands and their versatility we did not have time to attempt to devise Condor equivalents.

Both Series 20 and dBASE II support maximum record sizes of 1024 bytes, but Condor's Series 20 makes more efficient use of that space by representing numeric values in binary and by recognizing a date data-type which requires only 3 bytes. Series 20 also allows 127 fields to dBASE II's 32. dBASE II allows as many as 255 characters in any given field and, with its programming language and BASIC-like string functions can make much more effective use of the contents of string fields. The availability of a "date" data-type in Series 20 will be important in some applications involving calendar durations because tests on dates could easily be included in command parameters and expressions without concern for the stored date format.

The procedure which we devised for the checking account problem in the Series 20 article revealed several clumsy features of that system. It was not possible to do error checking for improper accounts at each entry. Lack of immediate feedback of possible erroneous entries causes the operator to have to resort his paperwork to pull out the relevant items to re-enter some. Since there is no indexing in Series 20 the entry files had to be sorted. With dBASE II a program could be written to resolve all errors and to insure proper balance at each entry before it is accepted into the database. Thus the time consuming printing of error records would be avoided. The separate runs through the files to detect file errors and to compute the balance would not be necessary. The lack of temporary variables in Series 20 caused us to have to jockey the debits and credits and deposits and checks around to get the final results in the right place. A program in dBASE II could handle the checks and deposits accumulation in variables and place them in the ledger upon completion of the run after the balance has been proofed. The dBASE II programming language is no harder to learn than BASIC and is considerably more versatile for applications than Series 20's submit procedures.

A function by function comparison reveals that some functions require different strategies to accomplish in each system. Each system will have certain tasks it can perform more effieciently than the other. The versatility of dBASE II, however, allows a wider choice of strategies to solving any given problem.

Review Summary:

Good points for use of this package:
It is powerful and versatile.

Bad points for use of this package:
It requires learning another programming language.

Recommendations and potential application:

A simple application involving large files might be implemented more efficiently using Condor's Series 20 database system. However, the versatility of dBASE II could be used to implement more efficient solutions to problems found in business applications. dBASE II is an excellent tool for an applications programmer. It is preferable to any BASIC for business applications and would be a better choice for someone new to computers who wishes to attempt implementation himself.

CROSS REFERENCE OF CONDOR COMMANDS TO DBASE II COMMANDS
REVISED 27 JUNE 81

This table provides an untested translation of approximate functional equivalents from CONDOR to dBASE II.

Condor command name, command syntax	dBASE2 description of dBASE II equivalent
APPEND database1 database2	USE database1 APPEND FROM database2
CHANGE database st field is newvalue	USE database REPLACE ALL field WITH <exp> {, ... }
COMBINE database1 database2	USE database1 SELECT SECONDARY USE database JOIN TO result ON <exp> FIELDS <field1, ... field32>
COMPARE database1 database2 {not}{matching} f1 f2 .. field32	USE database1 SELECT SECONDARY USE database2 JOIN TO result ON <exp> FIELDS <field1, ...field32>
COMPUTE database st fr = f1 opr f2 opr ... f31	USE database REPLACE ALL field WITH <exp> {....}
DATE {mm/dd/yy}	SET DATE TO 'mm/dd/yy' STORE DATE() TO variable
DEFINE database {*}	CREATE database
DESTROY database { OK }	DELETE FILE database

(continued next page)

DIC	{drv:}	lists data dictionary LIST FILES
EMPTY	database	USE database DELETE ALL
FORMAT	formfile{.frm }	In dBASE II forms are created within program structures
HELP	file{.hlp}	displays help messages on the screen done within program structure
JOIN	database1 database2 using field1, ... f32 [D]	USE database1 SELECT SECONDARY USE database2 JOIN TO result FIELDS <field list>
LIST	database {by field1 field2 ... field32}	USE database LIST <field list>
LOG	does ^c	RESET
MOVE	destination-db=source-db {OK}	USE source-db COPY TO destination-db
POST	database1 database2 {matching }k1,k2..{OPR f3,f4,..f32}	USE database1 UPDATE FROM database2 ON <key> {ADD<field list> } {REPLACE <field list>}
PRINT	database {by field1 field2 ... field32}	USE database SET PRINT ON LIST <field list>
PROJECT	database by field1, field2, ... field32	USE database COPY TO result FIELD <list>
READ	database {input-filet	USE database APPEND FROM input-file SDF {DELIMITED }
RENAME	newname=oldname	RENAME oldname TO newname
REORG	database {formfile.frmt	USE database SELECT SECONDARY USE formfile JOIN TO newfile FOR EOF {creates new structure } SELECT PRIMARY COPY TO newfile {FIELD <list> {FOR<exp> } or USE database COPY TO formfile {FIELD <list> }

RESTART	{ drv: }	RETURN
SAVE	new-name { OK }	saves result file to new-name file use RENAME or COPY
SELECT	database {where condition }	USE database COPY TO result FOR <exp>
SET	<function><status>	dBASE2 contains equivelant SET commands.
SORT	database by field1 field2 .... field32	USE database SORT ON <fieldn> TO result {ASCENDING}{DESCENDING } ..... SORT ON <field1> TO result { ASCENDING}{DESCENDING }
STAX	database by field1 field2 .... field32	USE database SUM field1 TO variiable, field2 .... field5 TO variable COUNT ALL FOR<exp> TO <variable> ? variable,...,variable /countvariable, etc.
SYS		returns to CP/M QUIT
TERM		assign terminal type INSTALL
TITLE	'text enclosed in literals or quotes',T,L,R,DATE	SET HEADING TO 'heading to print in quotes'or variable
UPDATE	database {where condition}	CHANGE FIELD { field1, field2,... field32}{FOR <exp>
WRITE	database {outfile} {switch option }	USE database COPY TO out file SDF{DELIMITED { WITH<delimiter> } }

(continued next page)

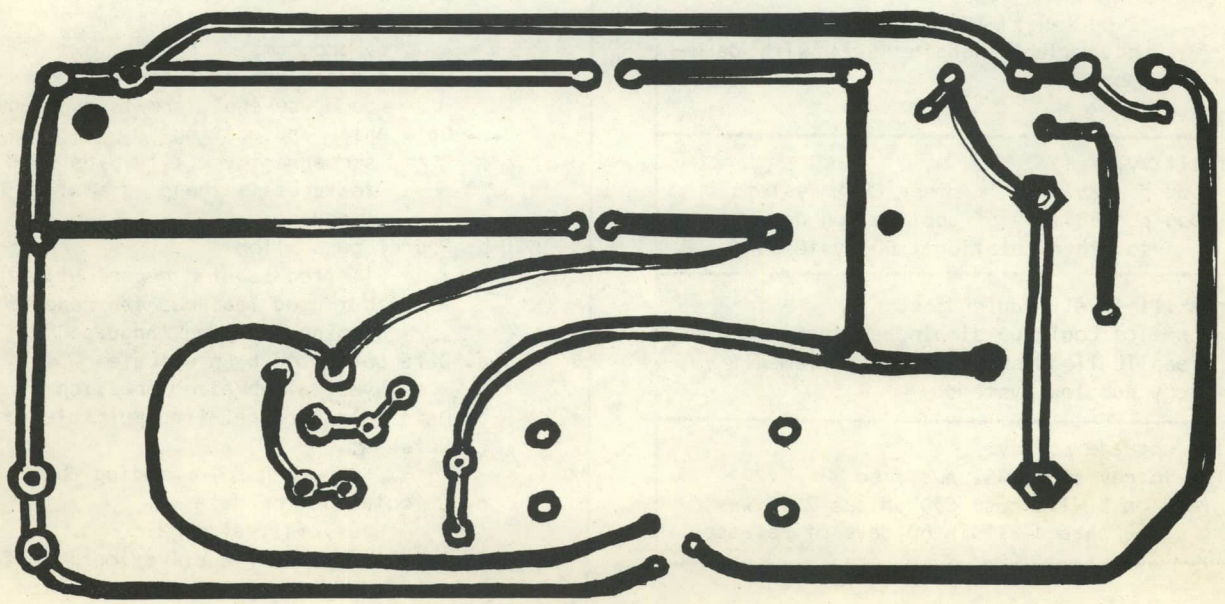


TABLE I
Facts & Figures

Package or Version name:
VERSION 2.02

Price:
$700

Systems available for:
CP/M, MP/M

Required supporting software:
A GOOD TEXT EDITOR
(an integral text editor for program creation will be added to a later revision)

Hardware requirements
8080 or Z80
Memory requirements:
48k plus bios requirements

Diskette capacity required:
2 drives with total capacity of >180k
(an application of normal size would require about 500k)

Utility programs provided:
(an integral programming language in addition to a full set of entry and reporting utilities.)

Record size & type limits:
fixed length records;
three data types are internally defined: character strings, integer and decimal numbers, and logical.
records are limited to 32 fields and 1000 characters
the database is limited to 65535 records. (i.e. 64k*128byte records = 8 meg, the CP/M limit)
strings are limited to 254 characters
numbers are limited to 10 digits
external record types are;
fixed length ASCII without delimiters, and variable length ASCII with delimiters

Portability:
good portability to other CP/M systems
good portibility of application designs to other relational DB systems

User skill level required:
a novice could do simple applications
a applications expert could implement very complex systems

System upgrade policy:
up to revision 2.02 are free
version 3 will cost $50 unless 2.02 was purchased within 60 days of release

TABLE II
Qualitative Factors

	Rating*
Documentation	
organization for learning	6
organization for reference	5
readability	6
includes all needed information	4
Ease of use	
initial start up	6
conversion of external data	7
application implementation	5
operator use	6
Error recovery	
from input error	7
restart from interruption	6
from data media damage	5
Support	
for initial start up	6
for system improvement	6

* Ratings in this table will be in a 1-7 scale where:
1 = clearly unacceptable for normal use
4 = good enough to serve for most situations
7 = excellent, powerful, or very easy depending on the category

TABLE III
Data Management Capabilities

A. Underlying Data Model
1. Data Types — char, integer, decimal, logical
2. Relationships — full n:m relations

B. Functions Provided
1.a. Data dictionary maintenance
no access to definitions by programs
b. Data reorganization & conversion
good, several formats available
2.a. Data entry and editing
screen entry utlity plus good features in the programming language
b. Report generation
limited tabular report utility, but good features for reports in the programming language
3.a. Data selection by predicate
yes, algebraic expression type
b. Data joining & relating multiple data sets
yes, with some weeding ability
c. Calculations on data
yes, full algebraic
4.a. Data independent application interface
only through data conversion

REFERENCE SECTION:

See the introductory articles for details on terminology and the evaluation format used. LIFELINES Volume 1, No.4, p.7; Volume 1, No. 5, pp. 4-6; Volume 1, No. 6, p.12

Send suggestions for software evaluations and other correspondence to:

The Software Evaluation Group
c/o Lifelines
1651 Third Ave.
New York, N.Y. 10028

Atre,S. Data Base: Structured Techniques for Design, Performance, and Management, John Wiley & Sons, New York,1980

Borkin, Sheldon A. Data Models: A Semantic Approach for Database Systems, The MIT Press, Cambridge, Mass., 1980

Date, C.J. An Introduction to Database Systems (2nd. edition), Addison-Wesley, Reading, Mass., 1977

Hutt,A.T.F. A Relational Data Base Management System, John Wiley & Sons, Chichester, 1979

Ullman, Jeffery D. Principles of Database Systems, Computer Science Press, Potomac, Maryland, 1980

Ed Paulette is Associate Executive Director of The Michigan Evaluation Resource Center, where he is engaged in the implementation and support of financial applications and client data base systems for human service organizations. His address at the Center is 338 South Sta St., Ann Arbor, MI 48104.

Steve Patchen is a data engineer who provides hardware and software support for small business computer users in the southeastern Michigan area. His address is: LAB Data Systems, 255 Chippewa, Pontiac, MI 48053.

OOPS!

We are sorry to report that there were several errors in ZOSO last month. (See this month's column, page 5.)

Page 3, column one, last paragraph, first sentence, should read: "Now usually I can cut tedious conversations short by pretending that I am about to be violently ill."

Page 4, column one, paragraph two, first sentence, should read: "Obviously, some 'Affordable' machines would belong in even lesser categories, but I have already written about threadbare computers, and in keeping with my policy of trying to write a class column, I refuse to discuss them any further."

Regrettably, an entire section was deleted, from page 3, column two, after the next to last paragraph:

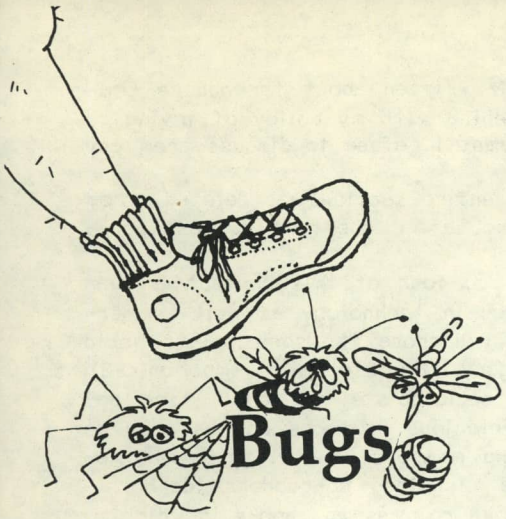
"The reason I'm so fond of this concept is that it ignores advances in technology as well as personal net worth. I suppose the sociologists would add a third category, The Stealables (mnemonic-S). I've never liked sociologists nor their lame notion that their felonious proteges act as they do because law-abiding members of society have somehow failed. These misguided do-gooders (do-horribles) have their own columns and books in which to snivel ad nauseam, so even if 'S' machines do exist, they will not be discussed here.

THE 'U' MACHINES

U1 MACHINES: Many computers are unaffordable, some more so than others. Generally Quite beyond the reach of Joe Average are those things the major nations (presently, the U.S.A. is one) use to process the 'final' algorithm for global obliteration. In this country, these billion-dollar beauties (let's call them U1 machines) are hidden inside some top-secret mountain, and are manned by two people seated some thirty feet apart. We may all rest easy because before any U1 machine will order the release of the 'Fifth Horseman', both of its operators must issue the same command within one or two seconds. This is called a 'fail-safe' procedure. Of course, there must be override capability, just in case one of these two guys turns bright blue and drops dead at his keyboard.

It should come as no surprise if I remind you that on occasion parts can fail (and U1 machines have many, many parts), backup systems can fail, humans who appear to be 'rock-solid' upstairs can lose it all with no warning whatever, and above all, the complex software which supervises the whole thing was written by a very large programming team. You know about those teams, right? Truly a case of too many cooks poisoning the proverbial broth, or if you'd prefer, subtle bugs which go undetected until everything blows up (if you catch the metaphor).

I've painted a grim portrait so far, and it gets even worse. On the other side of the world we have the dyspeptic Soviets. They too can 'write' the final chapter of human life as we know it. Somewhere in their vast land, there is a top-secret mountain under which they are doing more or less the same thing (as us). The rub is that they are doing it with much worse hardware. I presume this to be the case, because they keep trying to buy our U1 machines, and we keep refusing to sell them any. Also, from what I can gather, the state of software science in the Soviet Union is roughly on a par with their idea of 'haute couture' (high fashion - for the remedial study group). I always try to fill these columns with useful tips, and if nothing else, I hope I have given you new excuses to throw at your loved ones whenever they get on your case about alcohol or tranquilizer abuse."



BASIC Compiler
Manual
Microsoft's Basic Compiler has the ability to call assembly language subroutines, which may either be linked in with the 'COM' file, or they may be in a PROM. If you do it as specified in the manual (which works for the interpreter) the program won't link without errors. Here's how to do it. Use the call statement as below:

```
nnnn    CALL ADRES '
```

where nnnn is your line #. DO NOT define ADRES in the program. Your BASIC program should now be compiled. Now you define ADRES by writing a program for M-80 which has no instructions or data space as below:

```
;Program name could be CALLED.MAC
;
        public  ADRES
ADRES   equ     0F009H  ;  *
        end             ;  *
;
;
```

*Substitute the address of your routine here.

Now you can link the programs as below:

```
L80 BASPROG,CALLED,BASLIB/S,
BASPROG/N/E
```

(Note: The above should appear all on one line.)

BUG uBUG
Version 2.03
The current version will not load PLINK-II .PRG files (those with overlay information).

COBOL-80

DIVIDE (NUMERIC) by (LITERAL)

only works with GIVING statement appended, contrary to what is claimed in the COBOL-80 manual.

CP/M
Version 2.0
These bugs were contributed by Brad Barber of Mountain View, CA.

1-"Save 10" writes to file " . ". This file cannot be deleted or accessed from CCP.
2-ASM on a file without an end statement goes into an infinite loop. This contradicts the assembly manual on page 9: "The END statement is optional in an assembly language program."
4-CP/M function 11, get console status, returns a 01h if a character is ready. The Interface Guide says an OFFh is returned.
5-The CP/M Introductory Manual (rev. Jan. '78) page 11 states, "Further, CCP command lines can generally be up to 255 characters in length; they are not acted upon until the carriage return key is typed." This is incorrect; only 127 characters can be copied inot the default buffer area at BOOT+0080h.
6-The ED utility does not allow for form feed characters. Instead form feeds (control L) are always interpreted as a crlf. A literal control code would allow the user to override ED.
7-The DDT utility does not support return to CCP by a RET instruction. This feature is documented in the Interface Guide page 4. It can be supported by pushing a 0000h before calling the user's program.
8-CP/M function 13, reset disk system, resets the default disk id to A:. This creates problems for programs which should run on any drive while swapping diskettes. This can be repaired by using functions 25 and 14 around the reset call.
9-The PIP utility skips every other number when parameters "n2e" are used.
10-The PIP utility transfers only one line when parameters "s" and "q" are mixed. This failure is sensitive to search string size. 11-CP/M function 11, get console status, does not interact correctly with CP/M function 6, direct console I/O. After the first true get console status call, direct console I/O returns an ASCII null (00h) and the correct character is dropped. Later calls to console status/direct console I/O are processed correctly.

CP/M 28
Version 2.25
Magic Wand owners who update their CP/M to this version won't be able to use Magic Wand for TRS-80 until they get a new Magic Wand module for an ADM-3A or an ADM-31. The ADM-31 module will work better on Version 2.25 than that for the ADM-3A, but won't work at all with earlier releases of CP/M 28. The Magic Wand for ADM-3A should work in all cases.

Microsoft FORTRAN
Version 3.42
Two bugs have just surfaced in this version:

1-The first is the conversion of console output from lower to upper case. This will be fixed in version 3.43 which should be available soon.
2-The second is that in random disk I/O arrays will no longer be written or read from the disk with just a reference to the array name. Instead of the entire array being transferred, only one record gets sent. This will probably not be fixed in the near future, so use a DO loop to transfer all of the records.

MAC
Version 2.0
These bug reports also come from Brad Barber.

1-MAC produces an upper case listing even when the assembly source was in lower case.
2-MAC does not allow non-printable characters within assembly comments. For instance, users may want to include printer control characters such as formfeed.
3-MAC needs a true include statement as well as the MACLIB

statement. This is the same as MACLIB but with reading the file on both assembly passes. Includes are useful for modular program construction and for very long source files.
4-MAC flags "<>" as an error even though it is documented on page 10 of the macro assembler manual version 1980.
5-The instruction sbi should be flagged as a possible error. It is easily confused with sui (subtract immediate). Use of sbi in place of sui may not be detected during normal testing procedures.
6-If a program ends exactly at "END", the end-of-file mark (ASCII 1Ah) is ignored. If a carriage return follows END the file is processed correctly.

PASM
Version 1.02, with PLINK
PASM will assume that a common block not assigned any space does not exist. PLINK can give strange error messages when linking files produced by PASM containing empty common blocks. For example, assembling and linking the sample program on page 28 of the PLINK manual will produce "Pass 2 error 30" error message from PLINK, which indicates that the relocation base was undefined.

The fix is to assign some space to the .blnk. area with a ".blkb " pseudo-op. For example, the example program "Loader" on page 28 of the PLINK manual, write:

```
begin:   .loc .blnk.
         .blkb 1
```
instead of simply .loc .blnk.

Peachtree Accounting Packages
10-10-80

Accounts Receivable
Symptoms:
After doing an END OF PERIOD PROCESSING, the user will be unable to access his/her customer activity file.

Fix:
One sequence must be fixed in three programs. In the verify program, the line number is 220. In the fix and end-of-period-processing programs, it is line 60280.

The line reads:
FOR OP=2 TO EX%:GET FI%:

Add a comma and OP after FI%. The new line reads:
FOR OP=2 TO EX%:GET FI%,OP:

Inventory
Symptom:
Running the inventory status program gives a "SYNTAX ERROR IN LINE 54900" or line 2.

Fix:
Change the "TAB(2) in line 12100 to "TAB(20)".

Symptom:
Using the "MODIFY" command causes an error.

Fix:
Line 3250 should be:
RA%(FI%)=1:R$=MID$(RE$(FI%),1,18)
;:QL=18:GOSUB
7000: KE$(FI%)=R$:MO%=1:GOSUB
61200:IF ER%<>0 then 54100

Payroll
Symptom:
When "CALCULATE PAYROLL" is done, user gets basic error #9, a "SUBSCRIPT OUT OF RANGE" error in line 6230.

Fix:
In line 6220, "6220 NEXT" should read "6220 NEXT:i=i-1". In lines 6081 and line 8050 make this same addition.

PMATE
Version 2.06
When macros are defined for the permanent macro area, they are identified by labels. The manual states that any character or symbol can be identified by labels. The manual states that any character or symbol can be used for a label, but using any lowercase alpha character for a macro label in the definition will result in a "Permanent Macro Not Defined" error message when the macro is invoked. The solution is to use only upper case alpha characters or symbols, even though the macros can be invoked with either upper or lower case alpha characters.

ZDM/ZDMZ
Version 1.4/2.0
Under certain conditions it will write to memory during the two parameter form of the HEX/ASCII dump command. Fortunately, this form of the D command is rarely used. Orders mailed after 29 June, 1981 were corrected without a version number change.

ZSID
Version 1.4
We owe these bug reports as well to Brad Barber.

1-ZSID commands S and A should include a way to return to the previous location.
2-The A command should display the current contents before accepting user input.
3-ZSID supports one to fifteen character names while MAC can produce sixteen character names. If a program includes long names, the program's sumbol table can not be loaded by ZSID.
4-During progam trace with the T or U command, the user has only one chance to hit the rubout key. If he pushes any other key, the keyboard becomes inactive.
5-ZSID utility HIST.UTL dies if the trace range is too small.
6-The A command if given "cp f4" generates "cp04."
7-The A command if given "ld de 0fffa" generates "ld de 00fa".

A Note on MACRO-80

MACRO-80 Version 3.4 treats the concatenation operator, "&", a little differently than other assemblers, which can cause some confusion if you are used to another macro assembler. Most macro assemblers will print the concatenated label or word in the listing, but M80 prints an unconcatenated label or word. THe example in the M80 manual on page 24 produces not an "ERRORA" concatenated label, but an ERROR&A" label. You are still able to call or reference the label "ERRORA", and have the program supply the correct value, so this quirk should not be a big problem.

Random Numbers for Microsoft BASIC

by James R. Reinders

Microsoft BASIC needs a little help in coming up with a start for its random number sequence. For this purpose a RANDOMIZE statement is included in Microsoft BASIC and takes an expression as the seed for its random number generator. If no expression is specified, the program execution will be suspended and the user will be asked for a seed by the prompt:

Random Number Seed (-32768 to 32767) ?

I refuse to allow my programs to be marred by this question, nor will I accept the idea that the user has such control over the random number sequence. The same seeding at a program's start will provide the same sequence every time the program is run!

The question quickly becomes: Where can we get an unpredictable seed for the RANDOMIZE statement? Time of day clocks are nice but impractical since not everyone has a clock on their system (including me), and methods for getting the time vary from system to system.

Z80 enthusiasts look up! The Z80 chip has a timer on it! (Too bad if you don't have a Z80..but read on anyway...) The Z80 has a 7-bit counter known as the R register (Refresh). This register is incremented by one after every instruction fetch, the entire time the processor is on. This means that while BIOS, CP/M and all the rest await your commands this little counter is flying right along.

Zilog provided two instructions to access this register:

```
ED 5F    LD   A,R    ;put the contents of R into A
ED 4F    LD   R,A    ;put the contents of A into R
```

I have written two different routines to create a seed for the RANDOMIZE statement. One routine is for use in programs which will be compiled by BASCOM, the other is for programs which will be interpreted by MBASIC. The principles of both are the same...create an integer (two bytes) consisting of two values of register R--sampled at some random interval.

For programs to be compiled by BASCOM (into a .COM file for CP/M users) a small assembly language program will be created, assembled and later linked with a BASIC program. (See Figure 1)

The principle of this little routine is simple: take a two byte argument, discard the high byte and make the low byte the result's high byte and sample register R and use the value for the low byte of the result. This means that if the routine is called with an argument of zero, the result is the sampled value of register R plus 256. If called with an argument of one, then the result is the sampled value of register R. If called twice in a row with the same variable as the argument thaen the result is the first sampled value of register R multiplied by 256 plus the second sampled value of register R. Note: the argument given in the CALL must be an integer argument. Unpredictable results will occur if no arguments are given, if more than one argument is given, all but the first will be ignored.

Assemble the above program (on the disk as GETR.MAC) by typing:

```
A>M80 GETR,GETR=GETR
```

(This will create GETR.REL and GETR.PRN.) Now we'll use this in a simple program:

```
10 CALL GETR(X%)  :PRINT X%, :GOTO 10
20 END
```

To compile this (saved on disk as TEST.BAS) type the routine in Figure 2.

For programs to be interpreted by MBASIC, and using the same machine language program, I used what I think are some clever implementations of my own techniques. (See Figure 3)

GOSUB 60000 sets up a machine language program in a straing variable ASM$ using CHR$ to get each byte encoded into the string. The location of this string if found using VARPTR(ASM$). The VARPTR function returns the address of a three byte string definition. The first byte is the string's length and the next two bytes point to the string's start. This address is found and converted to a signed integer (line 60040) into the variable ADDR%.

A couple of simple alterations and this interpreter version could support the call as a USR function.

```
ADD: 60045 DEF USR=ADDR%
CHANGE: 10010 Q%=USR(Q%) :PRINT Q%, :GOTO 10010
```

Warning: I only intended for this routine to get a seed once, soon after program's outset. Be warned that is BASIC reorganizes the string space that the location of the string will probabaly change. I wqould recommend doing a FRE ("") to force reorganization and then a GOSUB 60020 to set up the pointers correctly if there has been a lot of string activity since last setting the address.

Using the routines is simple and very similar. (See Figure 4.) The first section can be separated from the second by various program initialization steps. I usually include an INPUT of the user's name. In the time it takes to type in your name, the timer counts over and over many, many times!

FIGURE 1

```
                        ; by James R. Reinders
                        ;   July 6, 1981
                        ;
                              ENTRY  GETR
                        ;
0000' ED  5F      GETR:   DB      0EDH,05FH       ; LD   A,R
0002' 46                  MOV     B,M             ; LD   B, (HL)
0003' 77                  MOV     M,A             ; LD   (HL),A
0004' 23                  INX     H               ; INC  HL
0005' 70                  MOV     M,B             ; LD   (HL),B
0006' C9                  RET                     ; RET
                        ;
                              END
```

FIGURE 2

```
A>BASCOM TEST,TEST=TEST  (this will create TEST.REL and TEST.PRN)
A>L80
*TEST                    (specify the file we are linking)
*GETR/S                  (Find GETR from GETR.REL)
*BASLIB/S                (Find BASIC routines from BASLIB.REL)
*TEST/N/E                (Exit and save as TEST.COM)
A>TEST                   (Run TEST.COM-resulting in an unending list
                                                        of numbers
```

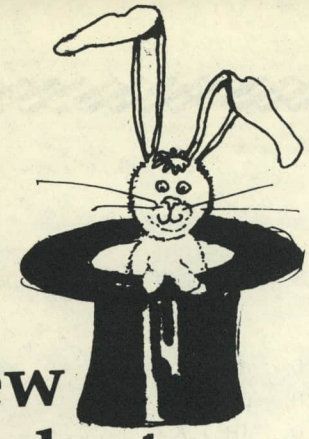
FIGURE 3

```
10000  GOSUB 60000
10010  CALL ADDR%(Q%)  :PRINT Q%,:GOTO 10010
60000  ASM$=CHR$(&HED)+CHR$(&H5E)+CHR$(&H46)+CHR$(&H77)+CHR$(&H23)
60010  ASM$=ASM$+CHR$(&H70)+CHR$(&HC9)
60020  ADDR%=VARPTR(ASM$)  :ADDR=ADDR%+1-65536!*(ADDR%<0)
60030  ADDR=PEEK(ADDR)+PEEK(ADDR+1)*256
60040  ADDR%=ADDR+65536!*(ADDR>32767)
60050  RETURN
```

FIGURE 4

```
Compiled Version      Interpreter Versions

CALL GETR(SEED%)      GOSUB 60000                GOSUB 60000
                      CALL ADDR%(SEED%)          SEED%=USR(0)
CALL GETR(SEED%)      CALL ADDR%(SEED%)          SEED%=USR(SEED%)
RANDOMIZE SEED%       RANDOMIZE SEED%            RANDOMIZE SEED%
```



New Products

Musical Applications of Microprocessors
by Hal Chamberlin
This book is intended to explain musical applications for 16-bit microprocessors, including chapters on analog and direct digital music synthesis principles. Waveforms charts, nomographs and sample control and generation programs written in BASIC permit the reader to experiment.

661 pp., $24.95

New Versions

BASIC Compiler
Version 5.3
The enhancements and changes represented by this new version are outlined briefly in Editorial Comments on page 2 of this issue. A more detailed appraisal will follow next month.

CBS
Version 1.3
The following changes are reported in this new version:

1-Sort step speed is increased when an "index to read file by" is used.
2-If the same program is used in successive steps within one job stream, program step chaining is eliminated.
3-"Hooks" for proper operation of THE FORMULA are now included.
4-The user number now appears on data inquiry screen along with the date and batch number.

MicroSpell
Version 4.2
These features have been altered or added in the new version of MicroSpell:

1-LEX files now can be spread over several disks. This means that it should work on limited disk space systems. Formerly 150-200K was required; now as little as 70K can be employed. However, more than one drive is recommended.
2-The auxiliary dictionary can be taken from any drive, instead of only the currently logged-in one.
3-The BUILD program can also work with a dictionary on several drives. To further conserve space, the user can now suppress the generation of back up files.
4-The INVERT program can read from and write to any disk.
5-A program called CUSTOM has been added to perform modifications which in the past required DDT.
6-For users with equipment which produces reverse video when the high bit of the ASCII character is set, highlighted target words are an option. The reverse video will not work on terminals requiring an escape sequence to enable it.
7-Some letter groups separated by a hyphen will be joined and considered to be one word for looking up: a) when the first group is followed immediately by a hyphen; b) when that hyphen following the first group is followed in turn by a carriage return, line feed sequence; c) and when that sequence or return is followed immediately by a second group of letters.
8-Now words ending in 'ic' will be rejected if the suffix '-ly' has been added.

PRISM/LMS, IMS, ADS
Version 2.0
PRISM/IMS and /ADS will include a new report generator with these features:

1-User definable report formats
2-Report headings and footings which may contain information from the data file
3-Nine levels of control breaks
4-Subtotals and grand totals
5-Calculations at detail, subtotal, and total time
6-Boolean record selection
7-A high speed assembler sort routine
8-A free form report format.

SELECTOR IV
Version 2.14
These problems have been corrected with the new version:

1-Unique key fields now cannot be changed, when recalling and editing records, to become duplicates of existing records--if the existing fields are not empty.
2-The automatic re-entry into data entry no longer occurs after a sort/merge and functions properly.
3-Now one of multiple duplicated key records can be deleted, re-entered, and subsequently recalled properly.

'@s' is now used in the Page Report Generator. This function resembles '@L###', '@C' or '@R###' and is used following one of those on a particular line number. It designates that the literal or field at hand be printed immediately to the right of the preceding field and separated by one space.

An enhanced date function permits the addition or subtraction of a number less than 731 and the automatic return of the new date. This function works for dates from 1912 to 1999. Leading zeroes now are significant when logging in the system date.

There is another change regarding the use of dates; a value of zero (0) is assumed to be the system date; whereas, no value at all (pressing RETURN) is treated as a null. If a date field is assigned the value of '0', then the system date will be entered. This holds true for all procedures.

T/MAKER II
Version 2.2.2
The following new features are incorporated into this version:

1-The Unload function now destroys the working file from the current line forward as stated in the manual. The previous version deleted only those lines from the top line on the screen to the last line Unloaded.
2-A possible situation in which the Compute and Combine function would consider a blank line as a line containing data has been eliminated. This situation is extremely difficult to produce, obvious when it happens, and corrects itself at the next pass, except for the zeroes, which may be put on the blank line.
3-The Tally function can now accept problems which require about 30% more storage space than the previous version.

XASMF8
Version 1.03
This new version corrects the following bug:

An error occurred in certain instructions if the operand, a scratchpad register address, was represented by an arithmetic expression whose first character was one of the letters "D", "I", "S". The special register symbols "D", "I", and "S" worked properly, however.

News Flash

CP/M 2.23 Single Density for North Star

This is the first release of the single density CP/M2 on North Star. The Lifeboat release number is 3 to be compatible with the double/quad density version of the same number.

1. Built in Horizon I/O supports all three ports. Printer or other device may use any port.
2. Ability to use other I/O with special CONFIG program. The system is now easily patched since it comes up immediately under North Star DOS using "GO CPM" which allows the use of CP/M programs.
3. SETCPM program allows inspection and change of most system options.
4. FORMAT and COPY programs are now included. You don't have to use DOS any more.
5. Automatic modification on any cold boot for a non standard address North Star controller board.
6. A single drive system may be used as a four drive logical system. CP/M will prompt you to switch disks.
7. Single density disk files are fully compatible on all Lifeboat CP/M systems, including older 1.4 disks and double/quad disks. Don't try to read double density disks however, the controller won't support it.

Rumor Has It . . .

.....that Tandy Corporation is rumored to be working on a computer using the Motorola M68000 16-bit chip. They are hoping to have this wrapped up for Christmas (?).....

Operating Systems

Description	Version
Apple II w/Microsoft BASIC	2.0
Cromemco System 3 8"	1.4
Cromemco System 3 8"	2.2
Datapoint 1550/2150 DD/SS	**2.2**
Datapoint 1550/2150 DD/DS	**2.2**
Durango F-85	2.23
Heath H8 with H17 Disk	1.43
H89 Heath/Zenith-Magnolia	2.2
iCOM 3812	1.41
iCOM 3712 w/Altair Console	1.41
iCOM 3712 w/IMSAI Console	1.41
iCOM Microfloppy(2411)	1.41
iCOM 4511/Pertec D3000 Hard	2.22
Intel MDS Single Density	2.2
Intel MDS 800/230 Double Density	2.2
MITS Altair 3202 Disk	1.41
Micropolis Mod I-All Consoles	1.4x
Micropolis Mod II-All Consoles	1.4x
Micropolis Mod I	2.20A
Micropolis Mod II	2.20A
Compal Micropolis Mod II	1.4
Exidy Sorcerer Micropolis Mod I	1.4x
Exidy Sorcerer Micropolis Mod II	1.4x
Vector MZ Micropolis Mod II	1.4x
Versatile 3B Micropolis Mod I	1.4x
Versatile 4 Micropolis Mod II	1.4x
North Star SD	1.41
Mostek MDX STD Bus	2.2
Sol North Star SD	1.41
North Star SD IMSAI SIO Console	1.41
North Star SD MITS SIO Console	1.41
North Star DD	1.45
North Star SD	**2.23**
North Star DD/QC w/Corvus	2.22
North Star DD/QD	2.22
Ohio Scientific	2.24
Ohio Scientific C-3B	**2.24a**
Ohio Scientific C-3C' (Prime)	**2.24a**
Processor Technology Helios II	1.41
by Lifeboat TRS-80 5 1/4" (Mod I)	1.41
by Lifeboat TRS-80 Mod II	**2.25A**
by Cybernetics TRS-80 Mod II	2.25

OASIS for Altos, Bell Controls, Billing, California Computer Systems, Compucorp, Cromemco, Delta, Digital Microsystems, Dynabyte, Godbout, GRI, Index Computer Systems, IBC Intertechnique, Kontron, Media Systems Corporation, Micromation Doubler, Morrow Thinker Toys, NNC Electronics, Onyx, Quay,S.D. Systems, Teletek, TRS-80 Model II, Vector Graphics, Vorimex, Zilog.

Hard Disk Modules

Description	Version
Corvus Module	2.0
Apple-Corvus Module	2.1
KONAN Phoenix Drive	1.8
Micropolis Microdisk	1.92
Pertec D3000/iCOM 4511	1.6
Tarbell	1.5

New products and **new versions** appear in boldface.

July 13, 1981

VERSION LIST

Product	S	M	OS	P	MR	$	
ACCESS-80	**1.0**		**CP/M**	**8080/Z80**	**54K**	**792**	
Accounts Payable/Cybernetics	3.1		CP/M	Z80	64K	500	Needs RM/COBOL
Accounts Payable/Structured Sys	1.3B		CP/M	8080	52K	840/40	
Accounts Payable/Peachtree	**7-13-81**		**CP/M**	**8080**	**48K**	**530/60**	**Needs OBASIC**
Accounts Receivable/Cybernetics	3.1		CP/M	Z80	64K	500	Needs RM/COBOL
Accounts Receivable/Peachtree	**7-13-81**		**CP/M**	**8080**	**48K**	**530/60**	**Needs OBASIC**
Accounts Receivable/Structured Sys	1.4C		CP/M	8080	56K	840/40	
Address Mngemt.Sys	1.0		CP/M	8080		150	Requires 2 drives
ALDS TRSDOS	3.38		TRSDOS		32K	80/35	
ALGOL/60	4.8C		CP/M	8080	24K	250	
ANALYST	2.0		CP/M	8080	52K	250/20	Needs OBASIC, QSORT or VSORT
APL/V80 Interpreter	3.2		CP/M	Z80	48K	500	Needs APL terminal
Automated Patient History	1.2		CP/M	8080	48K	175	
BASIC-80 Compiler	**5.3**	**5.24**	**CP/M**	**8080**	**48K**	**360/35**	
BASIC-80 Interpreter	5.2	5.2	CP/M	8080	40K	335/35	W/Vers. 4.51,5.2
BASIC Utility Disk	2.0	2.0	CP/M	8080	48K	75	
BSTAM Communication System	4.5	4.5	CP/M	8080	16K	200	
BSTMS	1.2	1.2	CP/M	8080	24K	200	
BUG/uBUG Debuggers	2.03		CP/M	Z80		129/25	Not for CDOS
BDS C Compiler	1.44	1.44T	CP/M	8080	32K	150/30	
Boss Finc'l Acctg Sys	**1.04**		**CP/M**	**8080**	**54K**	**2495**	**Needs 2 drive,min. 200K ea.**
Whitesmith's C Compiler	2.0		CP/M	8080	60K	630/30	
CBASIC2	2.07P	2.17P	CP/M	8080	32K	125/20	
CBS Applications Builder	**1.3**		**CP/M**	**8080**	**48K**	**395/40**	**CDOS version too**
CIS COBOL Standard	4.3,1		CP/M	8080	48K	850/50	
CIS COBOL Compact	3.46	3.46	CP/M	8080	32K	650/50	
FORMS 1 CIS COBOL Form Generator	1.06	1.06	CP/M	8080		150/20	
FORMS 2 CIS COBOL Form Generator	1.1,6	1.16	CP/M	8080		200/20	
COBOL-80 Compiler	4.01A	4.01A	CP/M	8080	48K	710/35	
COBOL-80 PLUS M/SORT	4.01		CP/M	8080	48K	845/65	
CONDOR	1.10		CP/M	Z80	48K	695/35	
CREAM (Real Estate Acct'ng)	2.3		CP/M	8080	64K	250	Needs CBASIC
DATASTAR Information Manager	1.101		CP/M	8080	48K	350/60	
Datebook	2.03		CP/M	8080	48K	295/30	
Dental Mngmt. System	**8.7**		**CP/M**	**8080/Z80**	**48K**	**750/NA**	**Requires CBASIC2**
DESPOOL Print Spooler	1.1A		CP/M	8080	19K	80	
DISILOG Z80 Disassembler	4.0	4.0	CP/M	Z80	24K	110	Zilog mnemonics
DISTEL Z80/8080 Disassembler	4.0		CP/M		24K	110	
EDIT Text Editor	2.06		CP/M	Z80		129/25	
EDIT-80 Text Editor	2.0		CP/M	8080		99/25	
ESQ-1	2.1A		CP/M	8080	48K	1495/50	Needs CBASIC2
FABS	2.4A		CP/M	8080	32K	195/25	
FILETRAN		1.2	TRSDOS		32K	99/20	1-way TRS-80 Mod I, TRSDOS-CP/M
FILETRAN		1.4	CP/M		32K	149/20	2-way TRS-80 Mod 1, TRSDOS & CP/M
FILETRAN	1.5		CP/M		32K	99/20	1-way TRS-80 Mod II, TRSDOS -CP/M
Financial Modeling System	2.0		CP/M		48K	300	
FORTRAN-80 Compiler	3.42	3.42	CP/M	8080	36K	435/35	
FORTRAN TRS	3.38		TRSDOS			80/35	
FORTRAN Package	3.38		TRSDOS			150/35	
FPL	2.0	2.0	CP/M	8080	48K	695/30	
General Ledger/Cybernetics	1.3C		CP/M	Z80	48K	500	Needs RM/COBOL
General Ledger/Peachtree	**7-13-81**		**CP/M**	**8080**	**48K**	**530/60**	**Needs OBASIC**
General Ledger/Structured Sys	1.4C		CP/M	8080	52K	840/40	
GLECTOR Accounting System	2.0		CP/M	8080	56K	350/25	Use w/CBASIC2, Selector 111 C-2
HDBS	1.04		CP/M	+	52K	300	
Lifeboat IBM/CPM	1.1		CP/M	8080		175	
Inventory/Peachtree	**7-13-81**		**CP/M**	**8080**	**48K**	**530/60**	**Needs OBASIC**
Inventory/Structured Sys	1.0C		CP/M	8080	52K	640/40	
JRT Pascal	1.4		CP/M	8080	56K	225/25	
LETTERIGHT Text Editor	1.1B		CP/M	8080	52K	200/25	
MAC	2.0		CP/M	8080	20K	120/25	
MACRO-80 MACRO Assembler Package	3.40	3.40	CP/M	8080/Z80		159/25	
Magic Wand	1.11		CP/M	8080	32K	395/40	
MAGSAM III	4.2		CP/M	8080	32K	145/25	For CBASIC/MBASIC
MAGSAM IV	1.1		CP/M	8080	32K	295/25	Needs CBASIC
MAILING ADDRESS Mail List System	**7-13-81**		**CP/M**	**8080**	**48K**	**530/60**	**Needs OBASIC**
Mail Merge	2.26		CP/M	8080		150/25	Needs same version Wordstar
MDBS	1.04		CP/M	+	48K	900/35	
MDBS-DRS	1.02		CP/M	+	52K	300	
MDBS-QRS	1.0		CP/M	+	52K	300	
MDBS-RTL	1.0		CP/M	+	52K	300	

The listed products are available from their authors, computer stores, software distributors, and software publishers

KEY

S Standard Version
M Modified Version
OS Operating System
P Processor
MR Memory Required
$ Price

These prices are given as approximations. Actual prices for these products will vary from vendor to vendor.
New products and **new versions** are listed in boldface.
\+ These products are available for Z80 or 8080, in the following host languages: BASCOM, COBOL-80, FORTRAN-80, PASCAL/M, PASCAL/Z, CIS COBOL, CBASIC, PL/1, and BASIC-80 5.xx.

VERSION LIST

Product	S	M	OS	P	MR	$	Notes
Microspell	**4.2**		**CP/M**	**8080**	**48K**	**249**	
Medical Mngemt. System	**8.7**		**CP/M**	**8080/Z80**	**48K**	**750/NA**	**Needs CBASIC2**
Microstat	**2.0**		**CP/M**	**8080/Z80**	**48K**	**250/NA**	**Needs BASIC-80,5.03 or above**
Mince	**2.5**		**CP/M**	**8080/Z80**	**56K**	**125/20**	
Mini-Wrehse Mngmt Sys	5.5		CP/M	8080	48K	650	
MP/M Operating System	1.1		MP/M	8080	32K	300/50	For Intel MDS 800
MSORT	4.01		CP/M	8080	48K	160/20	Needs COBOL-80
Mu LISP-80	2.03		CP/M	8080	24K	210/25	
Mu SIMP/Mu MATH Package	2.03		CP/M	8080	48K	260/30	muMATH 80
NAD Mail List System	3.0D		CP/M	8080	48K	115/25	
Nevada COBOL	1.404	1.404	CP/M	8080	32K	149/25	
Order Entry w/Inv			CP/M	Z80		500	Needs RM/COBOL
PAS-3 Medical	**1.76**		**CP/M**	**8080**	**56K**	**995/25**	**Needs CBASIC2**
PAS-3 Dental	1.62		CP/M	8080	56K	995/25	See above
PASM Assembler	1.02		CP/M	Z80		129/25	
Pascal/M	3.2		CP/M	8080	56K	175/25	Also for CDOS
Pascal/MT Compiler	3.2		CP/M	8080	32K	250/30	
Pascal/MT +	5.25		CP/M	8080	52K	500/30	For Z80 too
Pascal/Z Compiler	3.3		CP/M	Z80	56K	395/25	
Payroll/Cybernetics			CP/M	Z80		500	Need RM/COBOL
Payroll/Peachtree	**7-13-81**		**CP/M**	**8080**	**48K**	**530/60**	**Needs OBASIC**
Payroll/Structured Sys	1.0E		CP/M	8080	56K	840/40	
PL/1-80	1.3		CP/M	8080	48K	500	
PLINKII	1.08		CP/M	Z80		350	
PLINK Linking Loader	3.25P		CP/M	Z80		129/25	
PMATE	2.06		CP/M	8080	32K	195	
POSTMASTER Mail List System	3.3	3.3	CP/M	8080	48K	150/20	Needs CBASIC
Professional Time Acctg	3.11a		CP/M	8080	48K	595/30	Needs CBASIC2
Property Manager	**7-13-81**		**CP/M**	**8080**	**48K**	**925/60**	**Needs OBASIC**
PSORT	1.1		CP/M	8080		100	
QSORT Sort Program	2.0		CP/M	8080	48K	100	
RAID	4.7.3A	4.7.3	CP/M	8080	28K	250/25	
Real Est Aquisition Prog	2.1		CP/M	8080	56K	500	Needs CBASIC
Residential Prop Mngemt Sys	1.0		CP/M	Z80	48K	650	
Lifeboat RECLAIM Verification Prog	2.1		CP/M	8080		80	
SBASIC	5.4		CP/M	8080		295/35	
SELECTOR III-C2 Data Manager	3.24		CP/M	8080	48K	295/25	Needs CBASIC
SELECTOR IV	**2.14**		**CP/M**	**8080**	**52K**	**550/35**	**Needs CBASIC**
Shortax	**1.2**		**CP/M**	**Z80**	**48K**	**500/15**	**TRSDOS,MDOS too,needs BASIC-80 5.0**
SID Symbolic Debugger	1.4		CP/M	8080		120/25	N/A-Superbr'n
SMAL/80 Programming Sys	3.0		CP/M	8080		75/25	For CP/M 1.x
STATPAK	1.2	1.2	CP/M	8080		495/30	Needs BASIC-80 4.2 or above
STRING BIT	1.02	1.02	CP/M	8080		75/25	
STRING/80 bit	1.22		CP/M	8080		95/25	
STRING/80 bit Source	1.22		CP/M	8080		295	
Supersort I Sort Package	1.5		CP/M	8080		225/40	Max.record = 4096 bytes
T/MAKER II Data Calculator	**2.2.2**		**CP/M**	**8080**	**48K**	**275/25**	
T/MAKER II Demo	2.2.1		CP/M	8080	48K	50	
TEX Text Formatter	1.1		CP/M	8080	36K	105/50	
TEXTWRITER-III	3.6	3.6	CP/M	8080	32K	125/20	
TINY C Interpreter	800102C		CP/M	8080		105/50	
TINY C II Compiler	800201		CP/M	8080		250/50	
TRS-80 Customization Disk	1.2		CP/M	8080		75	
ULTRASORT II	4.1A		CP/M	8080	48K	195/25	
Lifeboat Unlock	1.3		CP/M	8080		95	Use w/BASIC-80 5.2 or above
Visicalc	1.37		Apple	8080	32K	150	
Wordindex	3.0		CP/M	8080	48K	195	Needs WordStar
WordMaster	1.07A		CP/M	8080	40K	145/40	
WordStar	2.26	2.20A	CP/M	8080	48K	445/60	
WordStar w/MailMerge	2.26		CP/M	8080	48K	575/85	
XASM-05 Cross Assembler	1.04		CP/M	8080	48K	195/25	
XASM-09 Cross Assembler	1.05		CP/M	8080	48K	200/25	
XASM-51 Cross Assembler	1.07		CP/M	8080	48K	200/25	
XASM-F8 Cross Assembler	**1.03**		**CP/M**	**8080**	**48K**	**200/25**	
XASM-400 Cross Assembler	1.02		CP/M	8080	48K	200/25	
XASM-18 Cross Assembler	1.40		CP/M	8080	48K	200/25	
XASM-48 Cross Assembler	1.60		CP/M	8080	48K	200/25	
XASM-65 Cross Assembler	1.96		CP/M	8080	48K	200/25	
XASM-68 Cross Assembler	1.99		CP/M	8080	48K	200/25	
XMACRO-86 Cross Assembler	3.40		CP/M	8080	48K	275/25	
XYBASIC Interpreter Extended	2.11		CP/M	8080		450/25	
XYBASIC Interpreter Extended CP/M	2.11		CP/M	8080		550/25	
XYBASIC Interpreter Extended COMP	2.0		CP/M	8080		450/25	
XYBASIC Interpreter Extended ROM	2.1		CP/M	8080		450/25	
XYBASIC Interpreter Integer	1.7		CP/M	8080		350/25	
XYBASIC Interpreter Integer COMP	2.0		CP/M	8080		350/25	
XYBASIC Interpreter Integer ROM	1.7		CP/M	8080		350/25	
Z80 Development Package	3.5		CP/M	Z80		130	N/A Suprbrn,Magnolia,mod CP/M
ZDM/ZDMZ Debugger	1.2/2.0		CP/M	Z80		45	For Micropolis,N'Star,Apple,IBM 8"
ZDMZ Debugger	2.0		CP/M	Z80		45	See note above
ZDT Z80 Debugger	1.41	1.41	CP/M	Z80		50	N/A Superbr'n,mod CP/M
ZSID Z80 Debugger	1.4		CP/M	Z80		130	See note above.

T/Maker II:™
it not only does more than VisiCalc,™ it does it on your computer.

VisiCalc is a fine aid for the computation of numerical problems. But it does have two major limitations: it is available only for a small number of systems, and its use is limited strictly to numbers, not words. To overcome these substantial limitations, Lifeboat Associates introduces T/Maker II.

Unlike VisiCalc, T/Maker II is designed to run on most small business computers with CP/M® or similar operating systems and a video terminal with cursor addressing capabilities. And soon there will be T/Maker II versions available for UNIX,™ RT-11™ and other systems.

Works with words as well as numbers. Like VisiCalc, T/Maker II reduces the manual tasks involved in computing and calculating financial documents. But since most business problems and reports involve words as well as numbers, T/Maker II also functions as a full-screen text editor for word processing.

T/Maker II is the most advanced aid for the analysis and presentation of numerical data and text material. In a matter of minutes, an entire document—including all edited text, all figures and all calculations—can be created, reviewed on your screen and reported in printed form.

T/Maker II turns your small business computer into a powerful, sophisticated and convenient tool. A tool that will save you money, time and energy, and eliminate the need for costly time-sharing.

With T/Maker II you can easily perform an unlimited number of analytical and reporting tasks which integrate numerical and text processing. You'll find T/Maker II perfect for such things as:

- Financial Statements
- Statistics
- Profitability Reports
- Revenue and Expense Analyses
- Portfolio Evaluations
- Price Lists
- Rate Structures
- Expense Accounts
- Cash Flow Projections
- Checking Account Reconciliations

...and much, much more.

Easy to learn and use. You don't have to be a programmer to operate T/Maker II. Just follow T/Maker II's easily understood and ordered instructions, set up your data in rows and columns, define the relationships and T/Maker II will do the rest: it will perform the computations and formatting necessary to prepare your document. When you're finished you can analyze your report on your screen or store it on a diskette. Or, you can have the report printed with presentation quality.

And when any changes have to be made, simply enter the new figure or relationship and tell T/Maker II to adjust and recalculate all the new results.

Editing capabilities. As a full-screen editor for word processing, T/Maker II handles text up to 255 characters wide. It includes features like text formatting and justification, centered titles, a text buffer for block moves and repeated inserts, global search and replace commands for printing your letters, reports and documents. Wide documents are supported by horizontal scrolling.

Low cost. The cost of T/Maker II is only $275 plus shipping and handling. Dollars well spent once you consider all the time, energy and money it can save. T/Maker II is brought to you exclusively and supported completely by Lifeboat Associates, world's largest computer software publisher. For more information send us the coupon below.

Mail coupon to: Lifeboat Associates, 1651 Third Ave., NY, NY 10028. Or call (212) 860-0300.

☐ Please send me more information on T/Maker II.

☐ Please send me a free Lifeboat 0 Catalog featuring over 200 programs, 5 including integrated accounting 0 and professional practice systems, 3 office tools for bookkeepers and secretaries and sophisticated tools for programmers.

Name_____

Title_____

Company_____

Street_____

City_____

State_____ Zip_____

T/Maker II is a trademark of P. Roizen. CP/M is a trademark of Digital Research, Inc. VisiCalc is a trademark of PersonalSoftware, Inc. UNIX is a trademark of Bell Laboratories. RT-11 is a trademark of Digital Equipment Corp.

As an example of what T/Maker II can do, see the chart below. The operator entered only the data shown in boldface. T/Maker II calculated and reported all the other values.

	— Actual —			Growth		Total	—Projected—		
	1978	1979	1980	Rate	Average	(000's)	1981	1982	1985
Item A	**42,323**	**51,891**	**65,123**	24.04	53,112	159.34	80,782	100,206	191,262
Item B	**45,671**	**46,128**	**49,088**	3.67	46,962	140.89	50,891	52,761	58,791
Total	87,994	98,019	114,211	13.93	100,075	300.22	131,673	152,966	250,053
% Item	48.10	52.94	57.02	8.88	52.69	158.1	61.35	65.51	76.49
% Item	51.90	47.06	42.98	-9.00	47.31	141.9	38.65	34.49	23.51
Total	100.00	100.00	100.00	—	100.00	300.0	100.00	100.00	100.00

LIFEBOAT WORLDWIDE offers you the world's largest library of software. Contact your nearest dealer or Lifeboat:

Lifeboat Associates
1651 Third Ave.
New York, N.Y. 10028
Tel: (212) 860-0300
Telex: 640693 (LBSOFT NYK)
TWX: 710-581-2524

Lifeboat, Inc.
OK Bldg., Suite 5F
1-2-8 Shiba Daimon
Minato-ku, Tokyo, 105 Japan
Telephone: 03-437-3901
Telex: 2423296 (LBJTYOJ)

Lifeboat Associates, Ltd.
PO Box 125
London WC2H 9LU, England
Tel: 01-836-9028
Telex: 893709 (LBSOFTG)

Lifeboat Associates GmbH
PO Box 168, Aegeristrasse 35
CH 6340 Baar, Switzerland
Tel: 042-31-2931
Telex: 865265 (MICO CH)

Intersoft GmbH
Schlossgartenweg 5
D-8045 Ismaning, W. Germany
Tel: 089-966-444
Telex: 5213643 (ISOFD)

Lifeboat Associates, SARL
10, Grande Rue Charles de Gaulle
92600 Asnieres, France
Tel: 1-733-08-04
Telex: 250303 (PUBLIC X PARIS)

Lifeboat Associates
Software with full support

LIFELINES®

1651 Third Avenue / New York, N.Y. 10028

NEW YORK N. Y.

FIRST CLASS MAIL
U. S. POSTAGE
PAID
Permit No. 416

FIRST CLASS MAIL