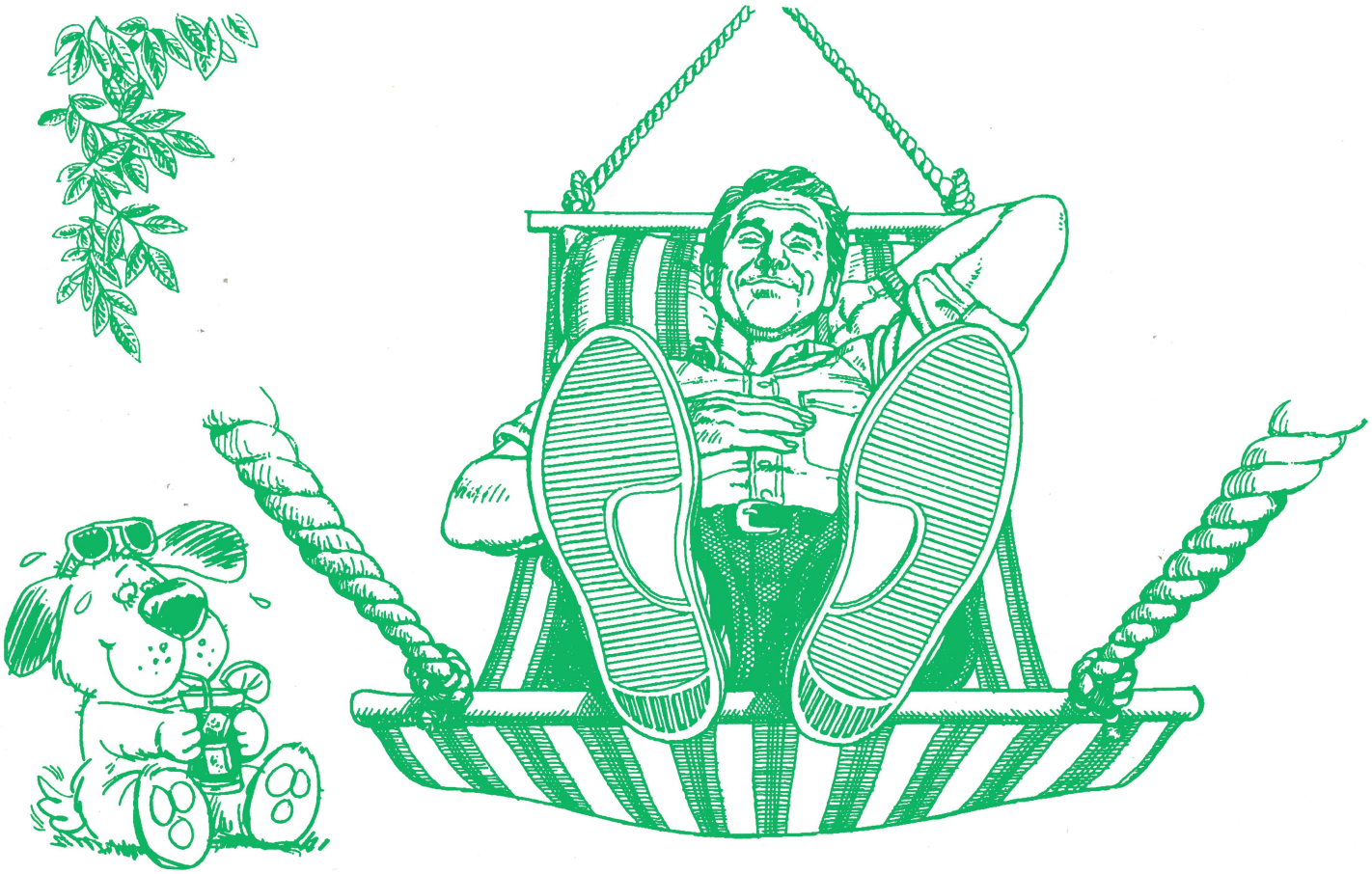
THE MISOSYS QUARTERLY

In this issue:

MISOSYS will be closed from August 8th through August 12th



Volume II.iv **$10** **Spring 1988**

TRS-80 Model I/III/4 Language Software from MISOSYS

Z80 RELocatable Assembler

MRAS [DOS 6.x M-21-083] [LDOS 5.x M20-083]

An advanced Z80 assembly package for the programmer who wants a powerful and flexible development system. It includes a macro assembler which generates either **relocatable object code** modules or CMD files directly, a linker, a librarian, a full-screen text editor, a utility for converting to/from line-numbered files, and a cross reference tool for directly generated CMD files.

MRAS generates M80 compatible /REL files. Supports REPT, IRP, and IRPC macros; nested includes, and a full range of nested conditionals. It has flexible output redirection of listing and symbol table.

MLINK supports virtual memory bit-stream buffering, REL and IRL library searching, zero disk space for DEFS in DSEGs and COMMONs, generation of program overlays, special link items: 0-3, 5-7, 9-11, 13-15.

Includes **MLIB**, our REL module librarian, and our **SAID** advanced **full screen text editor** which can be used to generate your assembler source code, C-language source code, or edit any type of ASCII file.

BASIC Compiler

EnhComp [DOS 6 M-21-072] [LDOS 5 M-20-072]

Released in 1986 and reviewed in the March 1987 issue of *80 Microcomputing* and October 1987 issue of *COMPUTER SHOPPER*. EnhComp has lots of great features; handles the bulk of Model III Microsoft BASIC and supports additional commands and functions. Standard is floating point with both **single and double precision** functions; random file access ("X" mode for reclens to 32767), turtle graphics, pixel graphics, keyed array sort, multi-lined functions, user commands, REPEAT-UNTIL, line labels, and more.

A *supervisor* program automates the edit-compile-test phases inherent when using compilers; this makes using EnhComp almost as easy to use as your BASIC interpreter.

With its built-in **Z80 assembler**, you can **easily** create hybrid programs of BASIC statements and in-line assembly code which **completely eliminate** contorted string packing and DATA statement high-memory module techniques for your BASIC program to access a machine code module.

You'll have to edit *existing* BASIC programs, but the power and completeness of EnhComp make that an easy task.

RATFOR Compiler

RATFOR-M4 #M-21-073

RATFOR compiles to FORTRAN; is **fully structured** facilitating modification and debugging; comments are simpler and more versatile than in FORTRAN, simplifying self-documentation. This allows changes without the subsequent debugging tolerated when modifying FORTRAN. RATFOR *compiles* source code to an *object* of FORTRAN; use your existing FORTRAN compiler to convert this to executable.

RATFOR is an excellent language for general purpose use, but it is vastly superior to FORTRAN when working with a large number of modules without documentation, as is necessary when producing very large programs.

Extensions supported include the "arith" macro to perform binary arithmetic operations, read and print macros for short form READ and PRINT, and support of any valid FORTRAN expression for "switch" and "case" operands.

This package includes the language translator, a *batch* file to automate compilation, a language Reference Manual, an Installation Manual, application programs in source code on disk, and our LED text editor for source code preparation.

Z80 Assembler

EDAS [DOS 6 M-21-082] [LDOS 5.x M-20-082]

This powerful combined disk-based **line editor** and Z80 **macro assembler** assembles from nested source files or memory buffer; nested conditionals with ten pseudo-ops, nested MACROs with parameters both positional and by keyword, cross reference listings; and a separate **full screen text editor**.

If you are writing system software, support software, applications - big or small, EDAS will provide the power to make your job easier, faster, and more worthwhile.

Z80 Disassembler

DSMBLR [DOS 6 M-31-053] [LDOS 5.x M-30-053]

Provides direct disassembly from CMD disk files, automatic partitioning of output disk files, data screening for non-code regions, and ***full label generation***. It even generates the ORGs and END statement - the complete ball of wax. You will find that the use of this disassembler - *even by a beginning assembly language programmer* - will be paying handsome rewards with the ease of its use and clarity of the documentation. It's a professional tool for your use.

The disassembler allows you to build a *screening data file* telling what segments of the program are to be interpreted as data regions. You enter the addresses of the "segments" after analyzing the target program's disassembly.

Output to **DISK** produces a file suitable for MRAS/EDAS and is automatically segmented into manageable file sizes.

REL Disassembler

UNREL [T80 M-30-054] [CPM M-32-054]

Decodes an M80-type relocatable object module and outputs an MRAS/M80 assembler source file. We bundle in **SPLITLIB** to split a library into separate modules and **DECODREL** to display the *bit stream* of a REL file.

UNREL assumes anything in a code segment is code, and anything in a data segment is data. It supports special link items: 0-3, 5-7, 9-11, 13-15.

UNREL should be the perfect professional assembler's tool for your bag of tricks.

Full C Compiler

MC [DOS 6.x M-21-064] [LDOS 5.x M-20-064]

If you are looking for a **full C compiler**, look no further. If you are looking for a well stocked UNIX System V standard library, look no further. **MC**, reviewed in the January 1987 issue of *80 MICROCOMPUTING*, is a complete C compiler which adheres to the standards established by Kernighan and Ritchie. The library of functions is extensive and System V compatible. The compiler generates Z80 relocatable macro assembler code (M80 or our MRAS). The libraries are files of relocatable object modules. **MC** is a full-featured compiler for the discriminating programmer!

MC supports command line *I/O redirection* for compiled programs, *wild-card* file specifications, parsing for UNIX "." extensions in file specifications, *overlay* support (requires MRAS), a full pre-processor, lots of options, and is designed for the programmer wishing the ultimate in C compilers. The package is supplied with the compiler, pre-processor, an optimizer, assembler macro files, C libraries, a Job Control Language file, the header files, and a 400+ page user manual. MC requires the use of either M-80 or MRAS (available separately), 2 disk drives, and upper/lower case.

Special Sale Prices

Product	Specify Model	Sale	S&H
DSMBLR	~~$34.95~~	$20.98	$2
EDAS	~~$74.95~~	$44.98	$4
EnhComp	~~$99.95~~	$59.98	$4
HartFORTH	~~$59.95~~	$35.98	$3
MC	~~$124.95~~	$74.98	$5
MRAS	~~$89.95~~	$53.98	$4
RATFOR	~~$99.95~~	$59.98	$5
UNREL	~~$49.95~~	$29.98	$2

Sale price is good through 8/31/88: You must include TMQ coupon. Add S&H. Check, M/C & VISA accepted. No COD's. 30-day refund on product if not acceptable.

MISOSYS, Inc.
PO Box 239
Sterling, VA 22170-0239

FORTH Compiler

HartFORTH [DOS 6 M-21-071][DOS 5 M-20-071]

HartFORTH is a ***full FORTH*** that conforms to the **79-STANDARD**. The Model I/III version is an indirect threaded version; the DOS 6 version is a direct threaded implementation providing greater execution speed of 10%-40% depending on the details of the actual program. The kernal contains some additional useful words and utilities which turn HartFORTH into a full-fledged development system.

HartFORTH is designed to ***run under an operating system*** which is totally transparent to the programmer or user. The virtual Memory that it accesses for storage and retrieval purposes is a normal DOS file that is requested by the FORTH system when it is first entered. Doing this has several advantages in that it provides for FORTH files to be used in other language application programs and vice-versa. Enhancements have been built into the kernal in the form of functions to call the operating system file handling routines so that other files may be created or accessed.

HartFORTH supports double length integers, string handling, cursor manipulation, graphics, random numbers, and floating point.

Take any ONE language product at 40% off!

THE MISOSYS QUARTERLY

Volume II.iv **Spring 1988**

Table of Contents

PO Box 239, Sterling, VA 22170-0239
703-450-4181

The Blurb by Roy Soltoff

Demise of 80 Microcomputing

Most of my readers probably know by now that 80 Micro has closed their doors. With the uproar generated over their elimination of 8-bit coverage in January, I won't spend much time here addressing the issue.

I would like to note its passing. Although 80 Micro may not have been held in our highest esteem throughout its tenure in the TRS-80 world, I must acknowledge that its presence did help create the industry of which I am a part. Friends aren't always in agreement; sometimes they argue bitterly. But in reflection, I try to examine the good things which friends have provided. I am saddened at the passing of 80 Micro. There were both good and bad things derived from it during its "reign". I shall remember the good.

TMQ's NEW look

In the last issue, our first printed with our NEC LC-890 printer, I commented that the issue was shot at 100%. Not so. Our printing company goofed and the entire issue was shot at 90%. This issue is definitely at 100% (I hope).

The issue was also generated mostly with a 10 point Times Roman font using 10pt leading. *Letters to the Editor* has some dialog concerning leading. This issue, after reading a bit on desktop publishing, was done with 11 point leading. Thus, it's probably even more readable than the last issue because of the additional leading (10/10 is a little too tight; 10/11 is better; 10/12 may be best, but more spacing means more paper means more weight means more mailing costs).

One of the books we read turned out to be an excellent introduction to desktop publishing and I recommend it to others. That's *Desktop Publishing Skills* by James Felici and Ted Nace. It's subtitle was, "A primer for Typesetting with Computers and Laser Printers". After reading and absorbing that book, you should have a firm foundation in all of the terms applicable to typesetting.

My reference to "LED Array printers", of which the NEC LC-890 is one, brought forth some queries, i.e. "What's a LED Array printer?". Here's an excerpt from the abovementioned book which should respond to those queries.

"Whether your perspective is that of the personal computer user accustomed to squinting over faded dot matrix printouts or of the publishing professional used to silver-film galleys, the laser printer is the defining factor of desktop publishing. Actually, many types of "laser printers" use imaging mechanisms other than lasers; in this book we go along with the convention of using laser printer as a generic term for any sort of plain paper raster-imaging printer with a resolution of 240 to 600 dpi. Higher resolution machines do play a role in desktop publishing.

Engineers refer to the laser as a "flying spot" imaging mechanism, because a single source of light moves back and forth rapidly across the drum. The laser itself does not move; rather a spinning mirror directs the beam across the drum.

As an alternative to lasers many manufacturers are using other light sources, including light emitting diodes (LED's), liquid crystal shutters (LCS's), and cathode ray tubes (CRT's). All these methods are described as *electrophotographic* because they produce a charge on the drum using some type of light source. A typical LED array incorporates about 2400 miniature LED's set in a row the width of the paper; each is independently controlled by the printer's internal computer. As the printer drum rotates, the LED's turn on and off up to 26,000 times per minute. A lens focuses the light onto the drum, which acquires a charge where the light strikes.

The LCS mechanism works in a similar manner. Thousands of individual liquid crystal shutters are set in an array. A light source is positioned behind the LCS array. When current is applied to the LCS, it becomes transparent; when current is removed, it becomes opaque. In this way the light beam is either allowed through to the printer drum or else blocked.

The advantage of LED or LCS array printers over printers that use lasers is that they have fewer moving parts. The disadvantage is that a single failed LED or LCS will cause streaking, which is particularly noticeable in graphics printing. The chance that any given LED or LCS will fail in a year's time is quite small, but the probability of at least one failure [in] an entire array of 2400 is much higher. If a malfunction does occur, most array printers are designed so that you can replace the entire array by removing several screws.

CRT printers are based on a single-line CRT. An electron beam flashes across the length of the CRT turning on pin

points of blue light, which activate dots on the print drum. An advantage of the CRT technology is that the coating material used on print drums is most receptive to the blue light that only CRT's offer which, means that less light is needed to activate dots on the drum and the drum's life is prolonged. Price has prevented the CRT from becoming a popular option. Quality is also a matter of question, since magnetic fields generated by the printer can distort the electronic beam inside the CRT.

Other processes have been developed which use methods other than light focused on a photosensitive drum. One such nonxerographic technology is *magnetographic* printing, in which a charge is written onto the print drum using methods similar to those used by the writing heads on disk drives. The magnetically charged dots pick up toner that has a slight iron content. One advantage of the method is that the image on the drum is more stable than with electrophotographic drums. A pattern can be built up on the drum in several passes, which means that the printer controller can build the image in sections and hence can get by with less memory.

Ion-deposition printers bombard the print drum with charged particles rather than light. The advantage is that the ion drum can be a steel surface, more durable than the light-sensitive drums or belts used by xerographic machines. Ion-deposition printers tend to be priced too high to be considered desktop publishing tools. Their forte is high-speed output; quality tends to be somewhat lower than that of laser printers.

Thermal-magneto-optical (TMO) printers work in a fashion similar to LCS array printers. A magnetic material, when heated, can be made rapidly either to block or transmit polarized light, according to an electronic signal. One TMO method, *electro-erosion*, requires the use of aluminized paper As the paper passes under a set of tiny electrodes, the printer vaporizes the shiny coating to create dots of black. This method has been used for 600-dpi printers, suitable for typesetting."

There you have it, and you thought there were only lasers! Now another topic is scanning. We have acquired an AST Turboscanner and OCR software. Actually, the AST is a Microtek MS300A 300dpi scanner. The OCR software is called *ReadRight*. It performs much better than we had expected it to perform since it does a credible job on dot matrix as well as the fonts it claimed to support. My guess on NLQ dot matrix is about 98-99% accuracy. Rather than attempt to explain what OCR software can do, at the conclusion of *The Blurb*, I'll print a page of sample text which OCR Systems provides. That sample scans perfectly - even with the mixture of fonts on a page.

We are also using the machine to scan graphics. To get some good graphics to import, we started a subscription to *Clipper*, a graphics service of Dynamic Graphics, Inc., a well known graphics company. Clipper is a monthly service which provides a portfolio of camera ready art. The company is the same one which puts out the Desktop Art packages for MS-DOS and MacIntosh. We made up this issue's cover from some Clipper graphics. The service is quite interesting. If we have the time, I'll add some of the little stuff throughout this issue. All that's necessary is to scan it into a .IMG file, strip out what I need and convert that to a Windows-PAINT file, then merge the picture into the WORD document file for printing with PageView. Because I'm getting down to the deadline with this issue, I don't know how much of that I'll get to. But it's fun to do.

MISOSYS Hot List

We have been requested to continue the MISOSYS hot list of products. This list represents what products in our catalogs have been the most popular in terms of sales units. The list excludes TMQ subscriptions and DISK NOTES. The current month is April 1988; prior 3 months is Jan-Mar 1988; and prior 12 months is Apr87-Mar88.

MISOSYS HOT List

Current Month	Prior 3 months	Prior 12 months
LDOS 5.3	LDOS 5.3	LDOS 5.3
PRO-WAM	PRO-WAM	PRO-WAM
RSHARD	diskDISK	Hrdwr Intfc Kit
diskDISK	*The Source*	Little Brother
The Source	RSHARD	PRO-DUCE
Hrdwr Intfc Kit	Little Brother	diskDISK

TMQ Schedule

Our target for mailing the *THE MISOSYS QUARTERLY* is the last week of the respective month as follows: Winter issue in February, Spring issue in May, Summer issue in August, and Fall issue in November. This schedule may place your TMQ late in the season based on the cover date; however, it follows from the mailing of Issue I.i on August 19th, 1986. We would like to pull up the mailing date about one month to place it in your hands closer to the first month of the season. But for now, the above periods are when it gets mailed. Please, if you have not received your issue by the last day of the following month, contact us. Also contact us with a revised address if you have moved - or are intending to move soon.

I have had some reports that it has been taking the USPS over **TWO MONTHS** to get TMQ to some of you via 3rd class. Although there is no guarantee of timely delivery, two months is ridiculous! Because I really don't know the exact date TMQ gets mailed when *The Blurb* is written, I can't state here the mailing date. However, based on past history, I predict it will be mailed on May 31st or sooner. The TMQ coupon has a field for entering your receipt date. We really would like to get some hard statistics as to when each of you receive this issue. Please note the receipt date on the coupon and get it back to us. Perhaps we can lodge a complaint with the USPS.

I also note that 3rd class postal rates went up over 25%; and 1st class as well as international mailing rates took a steep increase in excess of 17% as well. For instance, mailing a ten-ounce TMQ used to be $1.75, $3.58, and $4.72 for 1st class US, AO Air to Europe, and AO Air to Australia respectively. Now its $2.05, $4.22, and $5.55 respectively. I just can't see charging more for a subscription, so I'll just have to evaluate the impact over time. Perhaps advertising revenue can offset these increased costs.

TMQ advertising

Our TMQ subscribers have asked us to include advertising in TMQ from other companies who have TRS-80 products to sell. Since *80 Microcomputing* dropped TRS-80 coverage [and now stopped publishing], we've thought long and hard about this. Logical Systems used to accept outside advertising in their *Journal*; Lotus accepts advertising in their magazine - even from competitors. Why not MISOSYS?

So if you are interested in reaching a dedicated TRS-80 audience, why not consider *THE MISOSYS QUARTERLY*? If you have a TRS-80 Model 3 or 4 related product to sell, you can reach these buyers by placing your advertisement in our publication. TMQ is read world-wide. Our subscribers are predominantly in the United States; however, we do have a significant number in Canada, Europe, and Australia. Space rates are as follows:

Full page	$250
Half page	$150
Quarter page	$100

At this time, we are considering only black & white ads; however, we may consider accepting ads for our inside covers to be printed in the same color as the cover (TMQ alternates between PMS colors: green 354, purple 266, blue 293, and red 199). If you would like to place your ad in *THE MISOSYS QUARTERLY*, give me a call.

PD Software Librarian

Vic McClung has volunteered to be the librarian for the collection of TRS-80 public domain diskettes. Henceforth all requests and contributions be directed directly to him at:

Vic McClung
914 Crescent
Sikeston, MO 63801
USA

The XLR8er board

H. I. Tech wanted out. We wanted to continue to provide the XLR8er to our customers. So we took over the operation of distribution. Effective immediately, MISOSYS is the sole supplier of XLR8er boards. When you have a sole source, that usually implies a bigger price. Not with us! We lowered the price. Actually, what we did is absorb most of the price increase in 256K-150ns RAM which has occurred of late.

RAM prices have shot through the roof. What used to cost $2.75/chip now goes for $10-15, if you can get them. So we unbundled RAM from the XLR8er board and sell it for $175 w/o RAM. With 256K, we'll sell it for $245. However, we will not pay more than $9.50/chip for RAM; therefore, as long as RAM prices stay above that price level, we can only sell the board w/o RAM. I expect RAM prices to drop back to the $3.00 range by the 4th quarter of this year. The board will work without the extra RAM; you just won't have use of the RAMdrive. RAM can be added later just by plugging the chips into the sockets.

Here's a note on ordering. We have determined that a Revision A non-gate array machine will not work with the XLR8er board installed; thus if you have a Rev-A board, don't order an XLR8er. A 260-1069 is a non-gate array Model 4; the gate array is 260-1069A. That "A" is significant. The revision number is stenciled on the trace side of the motherboard. Rev A and possibly Rev B are no go; Rev C and Rev D are okay. On the other hand, if you have a 1069A, then that is okay, too.

A 4D is a gate-array machine with double-sided drives and one additional key on the keyboard. That is a BACKSPACE key which is hardwired to the LEFTARROW key. When Tandy shifted to the clustered arrow keyboard, they got many complaints about the use of the LEFT ARROW and its position on the keyboard for use as a BACKSPACE; thus the added key. Aside from that, there is no other difference between the 4D and the gate-array 4, to my knowledge.

THE SOURCE, 3-volume set

We still have a few hundred sets of the complete commented source code to TRSDOS 6.2 in book form excluding BASIC, hard disk support, and HELP facility. Here's our final closeout price: **$40 delivered**. We will soon clean house and throw this stuff out. If you want a set, get your request in. This offer is open to anyone; you do not have to be a TMQ subscriber. You can also buy as many sets as you want - until our supply is gone.

Other Open specials

Make note that from now until August 31, 1988, the following special deals are in effect:

1. Purchase a PRO-WAM Release 2 package at regular price and get a free Mister ED application pac.

2. Purchase a LB Data Manager at the regular price and get a free LB Maintenance Utility.

3. You may purchase any one MISOSYS language product at 40% off. The TMQ coupon must be submitted to qualify.

You do not have to be a TMQ subscriber to qualify for these specials, except that deal 3 requires submission of either the TMQ coupon or MISOSYS ad from our mid-May mailing.

Out of print TMQ's available

For out of print issues, we are providing back issues of *THE MISOSYS QUARTERLY* via copier reprint. The price is **$12.50 (new price)** plus $2.75 S&H in the U.S. and CANADA. For foreign zone D, the S&H rate is $5.50; zone E is $6.50 [**note the increase in S&H prices due to the latest round of postal increases**]. The price for regular back issues still in print is $10 + S&H. Currently out of print issues are include I.i, I.ii, I.iii, I.iv, and II.i. I am considering dropping the reprint of Volume I issues since it's really becoming a burden on time here at MISOSYS. For the time being, I've just upped the price after re-evaluating the time we spend to duplicate, assemble, punch, and bind a reprint, as well as the cost of bindings. Feedback on dropping Volume I reprints is encouraged. Here's a synopsis of past issues:

I.i A cc for MC; Add SETEOF to EnhComp; Change baud rates with SETBAUD; unlock protected BASIC programs; WinCalc: a PRO-WAM application.

I.ii BANKER - RAM bank control; BINHEX revisited;; Expose on LDOS 5.3; Fractals in FORTRAN; Model 4P BOOT ROM exposed; Split REL libraries with SPLITLIB.

I.iii Extended Memory access for LS-DOS; Upgrading Little Brother to LS-DOS 6.3; Converting Mail File Data to LB; a "CAT" for LDOS, "KILL" for LS-DOS.

I.iv UNDATE reverses DATECONV; 80x86 assembly language; Converting LDOS filters to LS-DOS; Previewing output from SCRIPSIT; Learn MORSE with CODE/BAS.

II.i David Hall on the 64180; Gary Phillips on XLR8 & 4P; Doug Tittle on sorting PRO-WAM data; WORD with DW II.

II.ii Extended DATE$ of Model I LDOS 5.1.4; Input SUBroutine for QuickBASIC; BASIC Interface to @EXMEM; HIRES Graphics for MC; Focus on speed.

II.iii Headline driver for I/III LDOS; XLR8er installation; CTL255 filter for PRO-WAM; FIXBANKS for your XLR8er; 4P boot ROM disassembled; C bit fields.

DISK NOTES 2.4

Each issue of THE MISOSYS QUARTERLY contains program listings, patch listings, and other references to files we have placed on DISK NOTES. DISK NOTES 2.4 corresponds to this issue of TMQ. If you want to obtain all of the patches and all of the listings, you may conveniently purchase a copy of DISK NOTES.

DISK NOTES is priced at $10 **Plus S&H**. The S&H charges are $2 for US, Canada, and Mexico, $3 elsewhere. If you purchase DISK NOTES with the coupon which accompanies this TMQ issue, you can save $2.50; the cost then being only $7.50 + S&H.

TMQ Index

Elmar Von Muralt came through with an index to TMQ back issues. Oddly, Elmar used the exact tool I would have done for the job - our LB Data Manager. Unfortunately, it came in too late to be included in this issue, however, I'll consider it for either the next issue or a special Disk Notes. I just wanted to get the word out so no one else duplicated his efforts.

Market Research

Finally, we get to a little bit of market research. MISOSYS is seriously designing an external hard drive package with the following specifications: 20 megabyte half height drive (Seagate ST225), PC-type Western Digital hard disk controller, MISOSYS designed host adaptor, standard quality drive case for two half height drives e/w 60-watt power supply and fan, device drivers for both LDOS (model III mode) and LS-DOS (model 4 mode) along with archive/restore utility, and our diskDISK software. Our target price is $495.00. The beauty of this configuration is that everything except the host adaptor is re-usable in an MS-DOS PC environment. Thus, if you can your TRS-80, you have a 20 Megabyte drive with controller totally reusable in a PC.

Any external hard drive requires FCC certification. Such a process is expensive. Therefore, MISOSYS is asking for your response of interest in such a product from us. I would also be interested in hearing about other products you would like us to offer you (and at what price). We do have dealer arrangements with a number of distributors. Therefore, if you are not in a hurry, I may be able to obtain a desired component for you at a reasonable price. Let me know.

HartFORTH now available

Since the demise of Molimerx, Ltd., I have signed a licensing agreement with Andrew Graham for publication of his HartFORTH compiler which we used to sell via an agreement with Molimerx. Note that the price of the Model I/III or 4 package has been lowered to $59.95. We also are testing out HartFORTH-86, an MS-DOS version of the compiler. We hope to announce that product in about three months.

Family Update

Yes, its that time again. I'm going to make this short as I'm running out of space. Here's the latest information on the Soltoff family...

Brenda broke her ankle on March 24th. It happened while walking! See, walking can be dangerous. Of course she was a little off balance as she was six months pregnant at the time. No, she twisted to the side to avoid falling on her tummy. Now when I received the call from the pre-school (she had just finished with her bus run and was at the school parking lot), I rushed over there with crutches in hand. See, I know to be prepared; the crutches were Brenda's used during her last mishap. Well to make a long story short, Brenda was on her back for almost two weeks in a temporary cast. The permanent cast was on for about a month. All's well now; the cast has been off for a few weeks. But it sure was an interesting six weeks. I had to take care of practically everything here - including being the sole means of transport for the family, and the business.

I had hoped to get TMQ out a few weeks earlier than this to avoid getting too much work close to the expected arrival date of our third and last "bundle of joy". Brenda's expected due date is June 12th. So if you call here any time in June and get my answering machine, you'll know why. That makes it a little busy these days since the last month of pregnancy brings more frequent medical appointments. We have childbirth refresher classes on Wednesday evenings; we already had our *hospital tour*; May 25th I have to take Stacey and Stefanie on the sibling hospital tour (Brenda gets to go to classes by herself); and June 1 we take over LS-DOS 6.3 sales and support.

We are also preparing for Stacey's 5th birthday party to be held a few weeks earlier this year. She'll be five on June 8th. Just to make sure that I don't have to rush Brenda off to deliver our new born during the party, we moved up the birthday party to May 22nd - also the day of our anniversary.

Both Stefanie and Stacey are in the drop stage. That's the stage where they drop everything on the floor whenever they are finished playing with something. I can't imagine how two little ones can create such a mess; is that the same everywhere? I don't recollect that kind of atmosphere when I was a child; but then I don't recollect having one tenth the amount of toys these two have. Is it just our memories which fail us?

Both kids are in pre-school where the primary activity seems to be art work. Each week they bring home lots of "paintings", glue pictures (that's where odds and ends are glued to paper), and various constructions made from paper plates, old egg cartons, paper cups, etc. We had been saving old egg cartons for months and amassed a few shopping bags full which we *donated* to the school. Stefanie showed up one day with a spider whose body was made from a piece of egg carton. I didn't even recognize the source until Brenda pointed that out. Good job!

After I left home about the age of 20, I lived in a number of apartments over the next ten years. Moving around like that, most mementoes of my youth, that were still in my possession, disappeared one by one. Today, I have few things left from my growing up years. So I decided that I would keep almost everything that my children amass. So far, I have about 3-4 large boxes of the artwork. And they have only been in school 3 years collectively. How long can I keep up that storage? At least I know they'll have more of their growing years when they get to be adults than I have now. Memories can be rewarding.

Letters to the Editor

MISOSYS catalog, etc.

Fm Charles A. Ainsworth: Roy,

He who whispers down a well
About the goods he has to sell,
Will never reap the golden dollars
Like he who shows them round and hollers!

It's now almost a year since last I saw a MISOSYS catalog; as you may recall, I'm a regular purchaser of your items. In addition, I am one of a group of friends who are very interested in computers, so on several occasions over the years, when someone asked my advice regarding an intended software purchase, I have gone to your catalog and made a recommendation which often resulted in that person ordering from you.

But there seems to be a vicious circle. Until the last time I placed an order with you, I noticed that you send out a copy of your catalog when shipping the goods. So, as long as someone continues to order, he keeps on getting a catalog; but when he stops ordering for a time, he doesn't get one. Now, it seems to me that a potential buyer without an updated catalog is less likely to buy than one who does have an updated copy.

The last time I saw a listing of your software was in April 1987, TMQ I.iii, but probably there have been changes since then.

We all know that catalogs cost money and that widespread and indiscriminate distribution can cost a lot. So here are a couple of suggestions.

TMQ would be an excellent place to list available goods and prices. Perhaps a complete catalog in each issue would take up too much space, but you might like to think of having an abbreviated listing. For example, DSM4 might be listed in a couple of lines as a fast disk-sorting utility capable of handling random-access files up to a length of nnnnn records and of selecting wanted parameters as part of the sort. I feel the short description is important; the complete absence of a description can leave one guessing.

For instance, in TMQ I.iii, page 9, you show FM86; an uninformed reader might not have the least idea what it is, but might be willing to purchase if he knew, or at least ask for further information or a catalog.

Then, a coupon could be added for requesting the complete catalog. To prevent the frivolous or idly curious from deluging you with requests for catalogs in this way, you might consider charging a nominal amount, such as one or two dollars, refundable on the next purchase or something like that.

There are several suppliers around who do this. And the product listing should also clearly appear in the index to TMQ.

Fm MISOSYS, Inc: Charles, Your ideas are good ones. In fact, you will have already noted that the last issue of TMQ contained our complete catalog of TRS-80 products. We may continue doing that at intervals. We have also considered mailing flyers to a portion of the folks in our database. We have approximately 20,000 names but only about 4,000 have purchased anything in the last 12-18 months. We may try periodic mailings to that subset. We are also trying a direct mailing to all registered LS-DOS 6.3 users to which we have invited other companies to participate. The response from the other vendors has been quite gratifying. This is probably a first in the TRS-80 industry - about a dozen vendors are sharing in an advertising-only flyer. Whether that kind of thing is repeated depends on how receptive the recipients are with their orders.

When MISOSYS first started, we did use a scheme where a catalog cost a buck; the $1 was credited against the next order from that catalog. That was an attempt to eliminate or minimize the abusive folks who checked off all the numbers in the reader service card every month. Most of the reader service replies we got from 80 Micro showed 20+, which meant that the reader checked off more than 20 numbers.

What we have done for the past few years was not use the reader service labels. Instead, we scan the name in our database and see if they have received our latest catalog. If so, we don't mail another one. If not, we flag the name for the next mailing and generate our own labels. We attempt to use bulk mail for most of that but that can wind up generating lengthy delays until receipt of the information. We usually mail 1st class for MS-DOS requests and TRS-80 requests where we receive an urgent plea. Of course, now that 80

Micro is gone, we need not be concerned with reader service requests for TRS-80 products.

50% trade-in

Fm Jon Roberts: Roy, Out of curiosity, does the 50% off for trade-ins apply to software only? After all I do have a sprinter 1. I wouldn't think you do since the xlr8er is only being sold by you, but I have to ask.

Fm MISOSYS, Inc: Software only. Besides, a sprinter is not an product equivalent to an XLR8er. Perhaps its time to elaborate. The 50% trade-in offer applies to any MISOSYS-published software product. What you trade in has to be an equivalent product. That means both machine environment and product function. We don't accept a trade of a Model I package for a Model 4 package; they're not equivalent. We will listen to a special request for trading in a specific Model III product for our Model 4 product where we have no equivalent Model III product available. For instance, we have accepted a PROFILE III for a Model 4 LB because we don't have a Model III LB package. Also, what you need to submit is manual and disk(s); binders you keep.

MISOSYS goes laser...

Fm Kevin R. Parris: Got my copy today (04-Mar), and it looks really nice. I like the type styles you have chosen. And it is much easier to read, for some odd reason, than I remember previous issues being (but I do not have one at hand to make direct comparison). One comment- maybe if the "boxes" were a bit smaller you could squeeze in even more of the valuable text that TMQ is so highly treasured for. Yea, the boxes look nice, but it's the content that I really pay for. NOTE: That is only a COMMENT, **NOT** a complaint.

Fm MISOSYS, Inc: We had a comment some time ago that continuous text made it difficult to discern where one thing ended and another began. I considered the idea and felt it had merit. Thus, about two issues ago, I started adding whitespace (about 3-4 column inches) at the head of each section. Then I had Brenda add the Merlin lettering. I felt that appearance was beneficial; thus, we continued that with the laser copy, albeit, we were able to drop the Merlin lettering and go straight from WORD with a 24pt font for the heading. I think most folks like it that way now, although you are the first to comment.

Fm Kevin R. Parris: I agree with the need to delimit the articles visually, but it just seemed to me that maybe the boxes were too much bigger than the words they contained. And let me repeat that I am not complaining. Also, I now have both the newest issue, and the Winter 1987 issue in hand together for comparison- the laser format is definitely much easier to read. I have never liked the "insert spaces to force flush right margins" that most newspapers (and you in previous issues) use. The proportional type is a MAJOR improvement, the way I see things. Keep up the good work!

Fm MISOSYS, Inc: The box just delimits the space. The alternative is to make the box smaller but precede and follow with blank space. I think I can afford to add a little blank space to dress up the division. Thanks for the input. A laser printer does add to the readability.

Fm Peter Schipelliti: I received the latest TMQ on Friday and I would also like to add that you did a splendid job. It is very readable, even at 1:30 AM with the little light behind the bookcase headboard of my bed. As usual, very informative and very well laid out. The type styles that you have chosen do their job well, and now I wish I could afford such a nice printer. Please continue the family update. It adds to TMQ something that most publications and newsletters don't even know about. The MISOSYS HOT List is a good idea. It looks like I have most everything from the current month except LB. Keep up the great work! Good Health to you and the family.

Fm MISOSYS, Inc: Thanks, We just may push to get the next issue out a little early to avoid conflicts with Brenda's expected delivery date. That's supposed to be June 12th. So I really don't want to schedule a lot of workload for June; and I don't want to be late with the Quarterly.

Fm Peter Schipelliti: It was nice to hear from you and I can understand the crazy schedule that June will bring. Perhaps you'll have a Flag Day delivery on June 14'th, And if its a girl you'll have to name her Betsy (Betsy Ross...). Best of luck to you and Brenda and thanks for your continuing support.

Fm Rex A. Basham: Roy, Just got the winter 88 issue of TMQ. Great stuff as usual! Almost went into cardiac arrest when I saw my name on the cover. Thank you very much for deeming my code relevant to publish. It was a tremendous boost to my quickly faltering ego.

Some feedback for you: The new format/print quality is superb; Very easy to read, no eye strain even after a 2 hour marathon session of devouring every page. On the PRO-MC 1.6 upgrade I have to agree with Pete Betz It would have been nice to stick the function sheets in the appropriate place of the original manual. I'm not complaining, the content is great and I'm very capable of operating a double to single sided copier.

JB and I both nearly split our sides upon reading the LSI Column ("What was on the disk before you started?"..."A green label" !?!?) And I thought I couldn't make a living as a consultant. In reality, I could probably triple my annual salary just answering questions like "which one is the enter key?". I'm surprised you haven't died of hysteria or frustration long before now.

I owe you an apology concerning the bench mark testing of the @MUL16 SVC. It's something I've been meaning to get around to but I haven't found the time. My real job at the phone company has been keeping me pretty busy and I've been doing some research in the hardware clock idea for the

4P. I'll try and get something together before the next issue is due out. Speaking of TMQ, I need the Volume I.i and my set is complete. There's a check in here for $12.50. Thanks a bunch for everything.

Fm Donald R. Arrowood: I'd like to say that you have really done a super job on this issue of the TMQ, it's like nice. Keep up the good work.

Fm LDOS Support: Speaking of the layout of TMQ, you mentioned in TMQ that you used ten point Roman - what leading (interline spacing) does Word use for that size type? It looked a little bit tight to my eye, was it eleven? Typographically speaking, twelve point leading is typically used for ten point type. In any case, this TMQ issue really looks great!

I know that this is a stupid, silly nit I'm picking, but I find I've become very aware of this kind of issue with the proliferation of DTP tools going out to the public in general. Then again, I can't believe the total junk that some folks are putting out on Imagewriters and bothering to duplicate en mass.

Fm MISOSYS, Inc: The interline spacing can be specified by point size. Thus, in places where I used 10-point fonts, I set the spacing to 10pt. Similarly, 8pt fonts (the listings) were set to 8pt spacing. The default is 12pt spacing for a 12pt font which is 1 line. Now the whole thing was a little tighter because contrary to the Blurb, my printing company did shoot the issue at 90% although it was supposed to be 100% per my specs. I did look at various inter-line spacing and thought I could get away with 10pt. It did give me a little more on a page.

Fm LDOS Support: Very interesting. Either Word isn't defining interline spacing the way that typographers do, or they're adding a fudge factor, or they don't know how to do it "right".

A typographer would use fourteen point leading for twelve point type, and write that spec as 12/14. Eight point type would be set as 8/9 and ten point at 10/12. The rule of thumb is one point extra if under ten point, two if over ten but under, say, eighteen.

Eighteen point and larger type is not typically used to set a body of text, but you would use 120% of the point size for leading, and then adjust it until it "looked right".

Leading is defined as the distance from baseline to baseline in succeeding lines of text. The result of using leading equal to the point size is that with some fonts you could end up with the decender of a character on the preceding line actually overlapping a capital letter on the subsequent line.

Now, the nice thing about having a Postscript printer is you can try 9/10 without having to muck about with downloading fonts. Smatter' fact, I have eight and ten point Roman in my HP soft font set, but if I wanted to try nine, I'd have to buy another set of fonts.

Fm MISOSYS, Inc: WORD supports 8, 10, 12, 14, 18, and 24 point in just about every one of the 12 type faces. The line spacing can be set to any quantity of points. You just need to set the spacing independent of the character size. Now line spacing is a paragraph measurement, the entire paragraph will be spaced accordingly.

Fm LDOS Support: Well, subsequent examination shows I'm all wet. I don't have a true comparator with a typographer's reticle, but the leading is indeed a lot closer to what it should be than what I though it was.

After compensating for the reduction to 90%, the leading is at least eleven points, and could be more, perhaps as much as eleven and a half points, close to the desired twelve. Again, without the proper tools I can't tell exactly.

That should certainly be adequate. Nonetheless, for some reason it still appears "tight" to me, moreso than I think the possible missing half-point should account for. Mysterious...

Fm MISOSYS, Inc: Well I am going out tonight to search for a book on Desktop publishing. Maybe after that, I'll be smarter.

Fm LDOS Support: Interesting that "adjustable" sizes don't seem to be supported. I believe that with Ventura Publisher (haven't tried it on a PostScript printer yet) I can have "anything".

Fm MISOSYS, Inc: Ah, but WORD is not Ventura publisher.

Fm Les Mikesell: Actually, the point sizes that WORD handles are a function of the printer drivers. If you have a postscript printer that can handle it, you could easily add additional point sizes by mucking around in the PRD file. It is very easy for postscript printers since the character widths scale for the same font in different sizes - you don't need to make up new width tables.

Fm MISOSYS, Inc: I meant to imply what point sizes WORD provides as delivered. Certainly with a Postscript printer any software which supports more point sizes, that is fully capable. I'll wait until I get either Venture or Page.

Fm Les Mikesell: I think WORD is right on the money in regard to point sizes. Setting 12 point line spacing gives the expected 6 lines per inch. Word does have a nifty feature of allowing line spacing to be "auto" and will size everything according to the largest character on a line. It does use minimal spacing in this mode, though - no leading. This probably could be arranged by tweaking the PRD file to make the font appear slightly taller than it actually is, at least under postscript where the application is responsible for all vertical

positioning. It is possible to add additional sizes to the defaults that word knows.

Fm LDOS Support: Les, Perhaps I misinterpreted Roy, but I thought he said that he was using ten point type and had set the line spacing to ten points also.

Fm LDOS Support: Right, Roy, but if it doesn't support scaleable fonts, it doesn't truly support PostScript, that was my question. In another reply Les indicated that you could do other sizes, though modifying the printer driver. This isn't that unusual, many packages that provide a degree of PostScript support aren't fully implemented, and require such driver changes. WordPerfect 4.2 also falls into this category. Ventura (version 1.1) does not, it has full PostScript support, as will the new WordPerfect 5.0 release. I'm sure that Word 4.1 or 5.0 with full PostScript support isn't far off.

Fm MISOSYS, Inc: I believe even Ventura should not be classed as having "full" Postscript support, according to your use of the word "full". Doesn't it support fonts only to 72 point? What does full mean? The flexibility I have with WORD's support of my printer is sufficient for me now. True, as I get more expert, I can obtain any number of additional fonts for download to my printer (WORD does support downloading). And yes, if I needed additional font or point capability then I would look for additional support. But it would be more helpful to tie down that word "full".

Fm LDOS Support: If you get PageMaker, make sure it's 3.0 or newer for the PC. Older version have major deficiencies that make the product very weak in comparison to Ventura. I haven't tried 3.0 myself, but I've been told that many of the weaknesses have been addressed.

Fm MISOSYS, Inc: We aren't in the market for either PageMaker or Ventura Publisher at this time since I see both of those packages to be a burden for TMQ preparation. I really don't want to do an 80-100 page magazine with them. I do need some more layout capability. Since I placed an order on Monday for a scanner and OCR software, we expect to be merging in graphics into the next issue. The OCR capability will speed up the process of getting letter text into an issue. I hate re-typing. As an aside, we are purchasing AST's scanner with Readright OCR software.

Fm LDOS Support: Could we be confusing more than one thing here? Les says that 12 point spacing will produce six lines per inch. You say that you used 10 point text with 10 point spacing, and my measurements show it to be somewhat larger than that. If Word added a fudge factor, then Les's numbers wouldn't work out right.

Could it be that there are actually three numbers we are discussing? (1) the point size of the type; (2) the interline spacing (within lines of a paragraph); (3) the interparagraph spacing (between two adjacent paragraphs).

#2 above is what I am referring to when I say "leading". For those unfamiliar with typesetting terms, that is pronounced "LED-ING" in reference to the metal (Pb), rather than the political verb.

Fm MISOSYS, Inc: Certainly there are three things, if not four things at play. WORD allows you to set the character size in points, the line size in points, the space which follows a paragraph, in points, and the space which precedes a paragraph in points. The latter 3 sizes may also be entered in inches, centimeters, or in lines (i.e. 0.5 lines would equate to 6 points). Now I pause to bring up my PC... Now I selected 10 point character size, 10 point line spacing, and 1 line of space following each paragraph (0 space preceding). As another example, I established parameters for the header box as follows: 2.5 L lines line spacing, 2 li following and 2 li preceding the paragraph. The characters were Helvetica 24 point bold italic. Am I making sense? As another example, the HIBANKS listing was printed in 8-pt Courier with 8-pt line spacing.

Fm LDOS Support: OK, guilty as charged. I was certainly being somewhat lax in the use of the term "full" when referring to PostScript support in the first place, and didn't define what I meant by it in any case.

With (some) further thought, I don't think that anything truly supports "full PostScript" except the printer itself, and perhaps not even that. Since PostScript is a language, rather than merely a set of defined functions and capabilities, you can write "programs" for PostScript to produce that no existing canned application could duplicate.

Let's describe "rudimentary" PostScript support as similar to what WordPerfect 4.2 and Word 4.0 support - access to fonts based on a pre-determined subset of fonts and configurations.

Then, "limited" PostScript support would add the ability to easily access any font in the printer, and provide automatic sizing as needed. Ventura 1.1 would then fall into this category, along with WordPerfect 5.0 and whatever the new version of Word will be.

"Advanced" PostScript support would include abilities beyond this, to directly manipulate the printer via PostScript itself. Ventura provides a limited capability for this via "embedded PostScript files".

Does that sound a little better?

Fm MISOSYS, Inc: Yes, sure does.

Fm Les Mikesell: Joe, as I understand it, the point size of a font includes a minimal amount of white space above and below the actual character. Thus printing 10 point characters with a 10 point line height is the usual thing to do. However, in large blocks of text, this minimal spacing looks a bit dense, and leading refers to the practice of increasing the line spacing

a bit. Word does not do this for you in the "auto" setting, but it does allow you to specify the line spacing for a block of text in any of several measures.

Fm LDOS Support: Les, that is not the case. I've forgotten the exact measurement spots, for some fonts which characters are used to measure point size are specified, but generally the point size of a font is specified by measuring the vertical distance from the top point of the highest ascender on any uppercase or lowercase character to the lowest decender on any lowercase character. The result is specified in points, as already mentioned.

Leading is measured also in points, and is typically measured from the baseline of a given line of text to the succeeding line of text. By definition, then, leading equal to the point size of the text would have zero whitespace between the bottom of the decenders on the preceding line and the top of the decenders on the following line.

Of course, the decenders and ascenders aren't going to touch that often, because the "right" combination of text would have to show up. Another consideration is that there is the equivalent of "artistic license" in designing fonts, and those who design fonts can and do sometimes design fonts where the ascenders and decenders a bit higher and lower than would be expected from the literal point size of the font in question. Whether or not this happens with PostScript or Bitstream fonts I know not.

Setting ten point type with ten points between adjacent baselines is **not** by any means the norm. Ask any professional typesetter or consult any decent manual on Desktop Publishing If that's what Microsoft says in the Word manual, they're wrong.

Les, Compare Courier (nominally twelve point) and Letter Gothic (also nominally twelve point). Which is taller? Letter Gothic, and by a bunch too. This is a good example of the variation in the height of what one might suppose were "similarly sized" fonts.

This comes from the good old days of typesetting, back when loose type was actually set from type cases. The point size was measured by the size of the base slug that carried the type character, and not the height of the character itself. This distinction has been slowly going away ever since the invention of the Linotype machine, but still remains as far as the distinctions between fonts go.

If you were setting type from individual pieces, you could indeed "set solid", that is, with no additional leading between rows. You then were relying on the base of the type to provide proper and sufficient spacing. By no means the norm, and it was generally done only with narrow columns and with certain typefaces.

S'matter fact, with some typefaces you can actually get away with "minus" or negative leading. Of course, you couldn't do this when working with real lead, this didn't come into play 'til optical (and now laser) typesetting.

Fm Shane Dawalt: Joe, on your definition of "point". Thanks, you anticipated my question. You said it was a measure of distance, what are the units? I mean, when someone says 72 point as opposed to 12 point. All I know is that one is bigger than the other. Is there any rule to go on that says 12 point is so high in millimeters or inches or whatever? Am I correct in that 72 point is LARGER than 12 point? I always thought it was the other way around ... but 72 is bigger than 12 so my thinking must be incorrect.

Fm MISOSYS, Inc: There are 72 points to an inch.

Fm LDOS Support: Yes Shane, 72 point is larger than 12 point. There are about 72 points in an inch, but it isn't exact. Wire sizes work the other way around, the bigger the number the smaller the wire.

How's the scanner coming?

Fm LDOS Support: Roy, I'd certainly like to hear how things go with that scanner. We've been looking (lusting?) at the HP ScanJet.

Fm MISOSYS, Inc: Well I expect to get it in quite soon. The price was too good to pass up. As you know, the AST scanner is pure MicroTek. That company makes about three different scanners - all received excellent ratings in PCM. AST uses the 300A scanner which is top of the line MicroTek. Differences between the scanners impact on intended use. For heavy graphics of all kinds of input, the flatbed scanner is preferred due to it accepting books, etc. The 300A is a autofeed page scanner. It is great for graphics and has a very good range of contrast controlled by a switch. The list for the AST is $2395. I was quoted $1399 by a mail order house. The ReadRight OCR software adds about another $500 to the price. I'll post my results after having had time to review the machinery.

Fm Shane Dawalt: I've been following this thread so that I may pick up some info (comments) about desktop publishing software. (I'm looking at the stuff ... but haven't decided what to get yet.) Out of curiosity, what is "OCR capability" in reference to the scanner?

Fm MISOSYS, Inc: OCR stands for ***Optical Character Recognition***. A scanner just is the equivalent of the front end of a copying machine. Instead of directly constructing a scanned image onto paper, it constructs it as bits. The scanner output is data file which holds the image scanned using some common format. There are a number of formats. TIFF, PIC, etc. Thus, if you scan a piece of text, you wind up with a data file that treats the text like a picture. That does zilch for input

to a text editor when you want the text to be a stream of characters.

The function of the OCR software is to take the bit image and make sense out of it, character-wise. So, for instance, when someone sends me a letter that I want to put into TMQ, I currently have to get the letter re-typed, unless it was provided on disk. Now, I hope to be able to scan the letter and pipe it right into WORD. That requires that the source document not be generated in a proportional character font as the cost of good scanners which handle proportional fonts are currently way out of the budget. Actually, its the cost of the OCR software, since the scanner is relatively static. Of course. most low-end scanners scan up to 300 dots per inch; which is equivalent to the laser printers resolution. Some high-end scanners which can accurately read text use up to 1500 bits per inch and some excellent software. We are spending under $2K for our scanning capability. That should suffice for most purposes. I told Brenda I'm liable to put in a scanned picture of the family on the next TMQ's family update section.

Fm Shane Dawalt: Ah HA, Roy. I though OCR had to do with those little bar codes. I couldn't figure out how OCR software would read a letter unless it was printed in bar code format. Now I see. And I didn't realize that the scanners had that sort of resolution. I suppose there is a limit as to how large text can be? I know it must be able to wholly fit on the scanner's surface. And that picture in TMQ sounds just dandy!

Fm MISOSYS, Inc: When the original is larger than the scanner's read surface, you just read it in segments. I also think the term OCR also applies to bar codes. The term probably handles a broad category of "character" recognition.

Fm LDOS Support: Shane, "OCR" stands for "Optical Character Recognition". That is, you feed in a document and it is converted to a bit map. Software then looks at this bit map and "translates" to text.

Nobody's got a perfect one yet, though the Kurzwell stuff is very good (like 98% or 99% accuracy, self-training, handles proportional fonts and so forth) but pricey ($10K for a full setup).

What you can handle with an inexpensive scanner and software (circa $2500) is not that bad, though you may be limited to monospaced fonts and perhaps 95% or better accuracy on "factory fonts", less on fonts learned in the field. That doesn't sound that bad, 'til you realize how many mistakes that means in a twenty page document.

Fm Shane Dawalt: 95% accuracy, huh Joe. Well, at least that's 95% of the document you'll not have to type!! I'm sure a spelling checker could catch the rest (and all proper nouns plus any acronyms ...).

Fm MISOSYS, Inc: Yes, That's still better than having to type that in (which still has problems in meeting 100% accuracy).

Fm LDOS Support: The problem is Shane, if that 5% is errors in source code, or columns of numbers, the amount of time spend finding and correcting errors is horrendous. Of course, the answer is that that's not the job that the $2K scanner is designed to do.

Fm Shane Dawalt: Ah Joe, you thought more ahead than I did. I was focusing on documents and letters only. Suppose one would leave that work to the $10K scanner!

Fm LDOS Support: Shane, You may be thinking of the special OCR-A and OCR-B fonts. These are fonts that were specifically designed for ease and reliability of recognition. For most books printed in the last few (three or so) years, the ISBN on the back is printed in OCR-A. In barcodes, if the number accompanies the bar code (in UPC anyway), it is usually printed in OCR-B. For 3-of-9, the number is in OCR-A. Both OCR-A and OCR-B are available on daisy printwheels, Spinwriter thimbles, Selectric golfballs and now in laser cartridge and soft fonts.

A nice thing about the Kurzwell is that it will do fully proportional type and doesn't much care about point sizes, so scanning from **any** commercially printed material is quite feasible. The $10K-$12K machine replaces equipment that didn't work as well, and was in the $40K-$50K category.

Rambling's on TMQ

Fm Fred Oberding: TMQ seems to get better with each issue. Your new layout is excellent - easier to read and re-locate articles. Got a good chuckle out of Bill S's LSI Column. His description of the less then intelligent Tandy customers had me rolling on the floor, and my wife wondering if I had gone crazy. Sorry you guys have to put up with it.

Fm Shane Dawalt: WOW!!! What a GREAT looking TMQ!! Laser printers are so nice (and high priced). I like the different point sizes, especially between normal text, programs & name headers on messages. VERY well layed out! [Enough ... or do you want more complements? <grin>]

Now for a question on TMQ, Winter 87. I finally got around to downloading the patches for Winter 87. There were PRO-MC patches I haven't installed yet (I know ... slower than molasses.). Anyway, after I downloaded them, I realized I didn't have a clue as to which patch was to be applied to which file. I assumed GETDEC60, ...DEC61 and ...DEC62 imply they are to be patched to MC/CMD, MC1/CMD and MC2/CMD respectively. Yes?? (Note, I am talking about PRO-MC only.)

Another patch, MCPUN6/FIX is to be patched to, I guess, PRO-MC. But to what, the preprocessor ... what?? Finally,

there's a patch to SAID (not PRO-SAID) with a filename of MRS610/FIX. Why is it '610' if it is not intended for PRO-SAID??

Please start labeling your patch files as you have in the past, with lines which state "Patch to PRO-MC, MC1" or "Patch to PRO-SAID" or something. This is quite confusing.

Fm MISOSYS, Inc: MRS indicates a patch to MRAS. The "6" refers to PRO-MRAS. The "10" indicates that it is patch number 10. I believe that each patch gives a comment line which states how to apply it via a PATCH command. Sometimes we do goof up in the designation. I'll try to be more careful of that in the future.

Fm LDOS Support: Actually, laser printers are cheap to buy, but fairly expensive to run (due to supplies and maintenance). For example, the HP LJ Series II is list priced at $2500, but can be found at around $1700. That compares very favorably to the list price of a NEC Spinwriter circa 1981 or so, something like $3500, or the introductory price on the RS WP-50 (replaced by the DW II before it was ever shipped in quantity, thank goodness).

A super duper printer like the one Roy has (the NEC LC-890, with Postscript) runs about $3200 or so discounted. For what it can do, that's a bargain.

Fm Shane Dawalt: Yeah, all patches do except for the ones in Winter 87 issue. But, with the info you gave me ... I'll do some more editing and patch my disks. Thanks for the fog lifter.

I just took a look at the patches in question. All patches except the MC/PRO-MC patches have the 'Apply via' statement.

One more question (well, maybe two) ... what is MCPUN6/FIX for? It has no 'Apply via' statement and I don't know what PRO-MC program to patch it to. I would guess MCP, but without editing and building the JCL then attempting the patch I don't know.

Second question: On the MRS => MRAS and 6 => PRO-MRAS, the patch, in Winter 87, says MRS610/FIX in the header, but under the 'Apply via' statement, the following is given: `PATCH MRAS MRS510`. Which is the typo, The 6 in the header (implying PRO-MRAS) or the 5 in the patch line (implying MRAS only)? I don't mean to get picky you understand. I'm just trying to get this cleared up before I apply these patches and stuff 'em in my patch archive.

Fm MISOSYS, Inc: The MCPUNx patches are for the preprocessor, as you assumed. I should have done a little better in clearing that up before printing. The MC patches are developed by Rich; he uses a different system for tracking them. My error.

The typo on the MRS610 fix was noted in a subsequent TMQ. The patch is for MRAS, not PRO-MRAS. The "Apply via,..." is correct. That crept in because I usually construct the V6 patch first, then re-edit the FIX file to construct the V5 patch. The "6" didn't get corrected to a "5". Of course, confusing the intended file to be patched is the significant reason for adapting LDOS PATCH utility to require the FIND lines for matching purposes as was integrated into LS-DOS PATCH facility.

TANDY CUSTOMERS

Fm Theodore Masterdon: I just finished the TMQ sections regarding the surprise emergence of the mass of "ignorant" TANDY customers that impacted on MISOSYS and Logical Systems in January. Fascinating. This confirmed something I have suspected for years.

I use two TRS computers today because in 1981 Radio Shack was the only dealer I could find who could sell me a computer AND a statistical package. The other stores were staffed by hacker-types who went on and on about technical details of the machines (which I could not understand and did not care about) and dealt with my statistical needs by stating that "there is a lot of software out there". But they didn't really know where.

The Radio Shack store kept me up and running, and had a lot of what I needed. I could call them when I needed help and get it. Later, I learned the limitations of my III and wished I had an Apple, but what the heck, my III did multiple correlations and multiple regressions and that was really what I needed.

Years later I was looking for a used Model 4 and spent many months calling for-sale ads. MOST of the people I talked to reminded me of me in '81, and your January 88 cohort. They treated there machines like refrigerators or adding machines,they used them for one application, and they rarely could tell me how much RAM they had, which operating system they were using, or even the proper names of the software they depended on. I often started out my inquiry with "what color is it".

So what's so bad about that? Tandy succeeded in marketing the world's first mindless who-cares machine. They sold them to one-purpose buyer's who felt no more need to learn the details of there DOS than they did to learn the details of their car's electronic fuel injection system or automatic transmission.

As a slightly advanced member of that species, I would opt for compassion rather than contempt. Let's face it, the difference between MISOSYS, or POWERSOFT, or LOGICAL SYSTEMS, and the Radio Shack is the difference between Carrol Shelby's racing garage and the local All American Main Street Ford dealer.

Fm Ray Pelzer: The sad part is that the description of a Tandy customer he gave is actually GENEROUS! Having been a Tandy employee, I can vouch for this. Having been a Customer Service Rep, I had first hand experience constantly! Two examples:

(1) A lady and her sons ran (maybe still do) a private investigations firm, and bought a Model 16b to run Xenix accounting and Scripsit-16. The woman refused to describe problems with valid English words... she made up words which she stuck to, and if I didn't use HER words, she refused to cooperate. F'rinstance, when I tried to tell her to sit down at the terminal and move the cursor down to the 5th prompt, I had to say "go to the CURSORARY and move the MODULAR down to the 5th line". Now, I bet you think I transposed the words, right? Wrong! To her, the 'cursorary' was the terminal itself, and the modular was the cursor!

(2) I got a call to go to a business on the west side which was having troubles with an old Model 2 and the 3-bay expansion drive cabinet... getting i/o errors during their programs. When I got there, I found out that they'd had the machine for about 3 years, and saw the disks looked badly worn out (oxide severely scraped at the directory track - common on mod 2's with heavy use at the time). I asked if I could see their backup disks, and they said they only had one set. Why? "Because these disks have worked fine since we got the computer." Further probing revealed that the salesman of the system made two sets of disks from the master at the time of the sale, and they'd used those ever since. Well, those babies were scored up too, and I asked if they'd noticed any tracks locking out when they reformatted the disks. They didn't, mainly because those disks had NEVER been reformatted before a backup since they got them... nobody told them to!

Need I go on?

Fm Shane Dawalt: Yeah, but the real 'headache' to ALL of this is that nobody wants to read the docs which comes with the hardware, software or whatever. I know a person who bought a calculator. Just an ordinary add/subtract/mult/divide calc. with memory and a simple little built-in printer. SHE couldn't even multiply with the thing. I asked her if she had read the instructions (I already knew the answer 'cause she doesn't read many instructions at all.) By golly, she hadn't. Where is logic when you need it??

And people who own computers should NEVER say they haven't read the manual. I mean, as Bill (and Roy) pointed out, some people didn't even know what a DIRectory was. How can you work without it??? I mean, hasn't those same people ever wondered to themselves, "Gee, I wonder if there's a command to get a list of the stuff I have on this disk?" Puts me the mind of a PASCAL class I took. A student wanted to read a data file into a program. The program kept aborting--it couldn't find the data file. Finally, she threw her hands up in the air for help. The assistant looked at the error message and asked her in which drive was the data file. She said, "It's on this disk [pointing to a disk still in it's sleeve]. Oh, you mean it has to be in the drive so the program can use it?" Now THAT was hard to believe.

And, as Bill and Roy both pointed out, they have had **many** calls like these. [How do people like this survive from day to day?]

Fm LDOS Support: Unfortunate but true. There are probably uncounted programs out there that present "results" that are approximate at best.

- numerical results produced from programs written by people who don't know what "significant figures" are - statistical results produced from "averaging" averages - spreadsheets accepted as gospel without the most basic of desk-checking - This all relates back to my comment as to what the availability of "desktop publishing" was doing to the average quality of documents and brochures. The easy access to computers and software has led to the general phenomena that many tasks previously accomplished by the "professional" are being taken over by the "interested (sometimes) layperson".

Is this a problem? Comments?

Fm Gary Phillips: Ted, I really think you are missing the point with your diatribe against hackers. I'm afraid there is no way you can argue that knowing the DIR command is a hacker capability. The real problem is that TANDY has failed to provide continued support for their limited-skill users. TANDY sold them the machines, and in many cases configured the disks, installed the hardware, provided very minimal training. Now that an upgrade is necessary, TANDY's reply is "Tough. Call someone else for help or buy a new MS-DOS system." The limited-skill user is directed to MISOSYS or LSI for assistance, and as you point out, neither of those organizations is equipped to support hundreds of folks who refuse to learn anything about their equipment.

Neither LSI nor MISOSYS is to blame for the fact that TANDY is now pretending that their Z80 customers no longer exist. The reality of economics should make it clear to you that customer hand-holding has to be limited when the product price is only $40. These irate users should be storming the towers in Ft. Worth with flaming torches, not tying up Roy's 800 number.

Believe me, I know what I'm talking about. Part of my job for seven years now has been providing support for this sort of user. And I'm quite willing to handle 6.3 or 5.3 installs in my area, for a reasonable fee. But at the going rate, it will cost these customers 2-3 times the price of the actual software to have it set up and put into operation for them. You can't demand that LSI or MISOSYS do that for free. Go yell at TANDY instead.

Fm Gary Phillips: Now, mind you Ted, I personally have little sympathy for this type of user. If you buy a car, and

refuse to learn to open the hood, or read the gauges, and refuse to learn about oil-changes, brake adjustments, and tune-ups (for a computer that's backups, disk drive cleaning, formatting, etc.) then eventually your car is going to quit working. At that point, you can go back to the dealer and throw a fit, and the dealer will offer to make the car work again, for some (probably large) fee.

You certainly have no right to go to the man you bought gasoline from (software for the computer) and insist that he fix your car for free. Chances are, he had been trying all along to tell you about oil changes and so forth, but you refused to listen. After all, the car was still running OK, wasn't it? He too may offer to get the car working again, but it is going to cost you extra beyond the price of gasoline. If you buy bargain gasoline, chances are this dealer doesn't even provide a repair (or in the computer situation, installation and software maintenance) service.

All analogies have weak points, but surely you see what I am trying to explain. Our society is full of people who depend on technology without the least comprehension of that dependency. When they get burned, they MUST take some of the responsibility upon themselves. My own rule is "If I can't understand it, and can't learn to understand it, then I can't afford to rely upon it exclusively unless I can pay the price for someone else to maintain it." This applies to cars, computers, appliances, or anything else.

Fm LDOS Support: Amen, Gary! And then there are those customers who know the value of knowledge and training, and are prepared to pay for it. Too bad their salesperson doesn't know what is available, and doesn't bother to sell it to them. That's my current problem.

Fm Theodore Masterton: Gary, I suppose things are always a little testy on a lifeboat. I am not taking issue with Mr. Soltoff's or anyone else's frustration with Tandy for misleading large numbers of non-hacker users by suggesting that MISOSYS or LOGICAL SYSTEMS can supply the hours of "hand-holding" that these users got from Tandy. My point was that I wanted to see a little more sympathy and less contempt. Tandy is the villain here, not some poor dope that is in a panic because his "no-worry" computer has something wrong with the date.

On that contempt issue; I have bitter memories of my early days being subjected to near ridicule by punk salesman who thought that, unless I shared their passion for the details of hardware I wasn't qualified to own and operate a computer. It made me mad then, and again now. Sure, a DIR command is about as simple as checking the oil in your Ford. But if the dealer told you all you had to do was drive it in and they would take care of it for you, who is to blame if I never learn how to check the oil?

Fm Shane Dawalt: One question, Theodore, how can you get by without the DIR command. I mean, doesn't someone (anyone owning a computer) wonder what's on a disk?

Fm Gary Phillips: Sorry, Theodore. I didn't mean to sound angry. But I don't mean to sound sympathetic either. If you call the dealer about a problem with your car, and he asks which model you are driving and you tell him "a white one," what kind of service can you expect? These users are, in my opinion, getting what they asked for in the first place. No, I don't expect everyone to learn how to open up the machine and rewire or repair it, or to patch damaged floppy diskettes, or anything like that. But basics like making backup copies and following extremely clear written instructions: yes, I expect it.

Fm Hardin Brothers: Shane, I have two friends who use M4's exclusively for word processing. Both happen to use LeScript, which has a built-in directory function, and so neither of them has ever tried the DOS DIR command. Their large brown TRSDOS notebooks just sit on the shelf gathering dust (both have learned how to use FORMAT, but neither has tried COPY).

Fm Shane Dawalt: Oh yes, Hardin. Forgot about those programs (few and far between) that display directories and such at the touch of a key (or by default). I suppose if that's all you use the computer for, and the software does the system level commands for you, then you aren't **forced** into learning the system level commands. And I can't imagine not using Copy! I've really started using COPY on my MSDOS machine! All my C source is copied onto floppy archival disks. Just the other day a C program (compiled on TurboC 1.5) nibbled at the drive D: FAT. CHKDSK found 5 clusters bad in 4 links. Turned out, it was nothing major. Basically, dead disk space had been reallocated. Next time, I will probably not be so luck. Ah, but the main backups are just an arms reach away!

Fm Jim Beard: My wife today observed a Tandy User in action (Reference: "The LSI Column", by Bill Schroeder, TMQ III.iii p. 30). The subject turned on a Model III with a 5 Meg LDOS 5.3 HD system boot disk locked firmly in :0 and got a screen of garbage. Whereupon, observing same, the subject began thumping the space bar rapidly, "pianoing" the keyboard with all 10 fingers, and popping the reset button with all four fingers of the right hand as if slapping a naughty child on the wrist. This was punctuated by a wide variety of other minor violence. The subject then announced "It's too hot, let it cool off for awhile" and turned off the main computer. After 30 seconds, the performance was repeated, this time followed by "It's broken!".

The last time I saw this machine up close, the HD and all available floppies had multifarious bullet holes, many hundreds of them, apparently from resets while disk accesses were in progress. At that time, I theorized that casual abuse of the reset button was the main cause, abetted by leaving the

boot disk in :0 with the door closed all the time. I don't get much satisfaction from knowing I was right, particularly since this is still going on.

Fm MISOSYS, Inc: Jim, That reminds me of my AT&T days. I used to be what was called a "Telegraph Serviceman". Fancy title for a testboardsman foxing out low-speed teletype channels. Anyway, we had a complaint from one of the military locations that their 28RO had a broken glass (that's like the plastic which covers up today's dot-matrix printers allowing you to see what is being typed). When the local TELCO repairman called in with the clearance report, he said it looked like somebody took a sledgehammer to the machine. Another irritated customer!

Fm Jim Beard: Roy, The Tandy User wasn't irritated. That was her normal mode of operating computers. She has absolutely no idea why she never has much luck. She wants me to teach her, but I'm too busy.

Fm Ken Stiefel: Roy Having finally gotten around to reading the Winter TMQ (not from lack of interest, just lack of time). I do have one question though. Did people really call up wanting the Upgrade asking for the WHITE Computer. I didn't think people came that dumb. Thanks again for turning out some excellent products, and please don't get so frustrated dealing with the screwballs of the world that you give up on the rest of us.

How Tandy sells computers.

Fm Charles A. Ainsworth: Roy, Probably I could tell you enough horror stories, along the lines of your correspondents, to fill an entire issue of TMQ, which, however, is better devoted to other things. But I could not refrain from telling you just a few.

But first, one of your correspondents mentioned that personnel turnover at the Tandy stores must be serious. I have personally witnessed this. I have been in my area five years, and have dealt with two stores; one I will call #1 and is nearby; the other, #2, is in a larger town up the road and is a full-blown computer store.

Store #1 has had four managers in the five years I have been around. It also employs two sales clerks and I have lost count of the number of changes in them, but I don't exaggerate when I say there have been at least fifteen different faces in those jobs in four or five years. Store #2 has had even more changes.

About two years ago, 80 Micro claimed that Tandy stores have an annual turnover of 90 percent of personnel. I am quite prepared to believe that.

It is obvious that no one anywhere along the line is willing to train these sales people in their products, and in the many times I have been in the stores, be it to buy batteries, flashlights, electronic accessories, computers, peripherals or whatever, no one around has ever been able or willing to give an intelligent and accurate answer to any of my questions.

My first story: About a couple of years ago, when model 4 gave way to the 4D, the outgoing 4 was reduced until it stood at :450; I ordered one. When I got the box home and opened it, here is what I found: The computer had a thick layer of dust on it, and when I opened it up there was another layer of dust inside which had filtered in through the ventilation grilles. There is usually a date handwritten inside on the base, which had been crudely obliterated but which I managed to read: April 1983, the earliest model 4. The machine had obviously been dropped and the CRT bezel was shattered in several places so the CRT flapped around in the breeze. The manual and diskette accompanying the computer were for TRSDOS 6.0.0 (at purchase date the current version was 6.2.1)

The disk drives were next to useless and sounded like coffee grinders and it was impossible to format or backup without a host of I/O errors. Obviously, this was a demo model that had had a hard time in some store.

When I took it back, the clerk asked "Well, and what did you expect for your measly $4S0, a diamond-studded machine?" He adamantly refused to accept it back until I threatened to send my lawyer around to start legal action.

Story W2: Sometime last January, the 4D was marked down to $600 in a flyer from Fort Worth, so I decided to try my luck again and went to store #1 to get one. Two sales clerks (new and unknown, as usual) definitely refused to take my order as they "weren't allowed to sell the 4", which they stated I could only buy at the full-blown computer store, #2. So off I went to store #2 and placed my order. The salesman at store #2 failed to understand why store #1 couldn't have sold it to me.

Recently, I was at store #1 which had refused to sell me the 4D and I commented to the manager what had happened, and he admitted that the clerks had been wrong and could have made the sale. However, he added that they preferred not to sell the 4D which doesn't comply with FCC regulations for RF interference. (!!!)

Story #3: Late last year, a friend purchased model 4 Visicalc from Tandy. It came with TRSDOS 6.0.1 and refuses to perform properly with any current DOS. The registration card was duly mailed, and several letters to Fort Worth have produced the usual: Nothing!

So, when shopping at a Tandy store, as the Romans used to say, *caveat emptor*. (That's Latin for Let the buyer beware); also, when getting information or advice, be prepared for mounds of, well... er... let's be reasonably polite and call it steer compost! And don't, under any circumstances, accept any verbal statements on any important feature. If what you want to know isn*t in the catalog, ask friends or read books and magazines or get information from anywhere but a Tandy

salesperson. It may be laborious but, in the long run, may save you time, money and ulcers.

The SOURCE, TMQ, Support, ...

Fm Michael Dauphin: Roy, I just returned from mailing another order to MISOSYS and found TMQ Vol II.III in my mailbox. Another excellent effort! I have already found another item to buy. Just as soon as this Msg is sent I'll be heading back to the post office with an order for the *THE SOURCE*. If you'd like you could hold up sending my original order until the check for *THE SOURCE* arrives in Sterling. (Should arrive the next day, it's after five pm. now) I won't mind the wait and that way you could ship both orders in the same package and keep a couple of extra bucks in your pocket and out of UPS's coffers.

The first order was for 1) LS-diskDISK and RSHARD. Would it be possible for you to throw the LDOS tabset and QRC into the box?

I read the thread in the TMQ which featured James J. Wood and his critical comments about software vendors and MISOSYS in particular. I feel his comments are way off base even though I share some of his frustrations. The responsibility for software purchases rests, as it should, with the BUYER. If someone buys software from a company with an unknown or poor reputation without knowing ahead of time whether or not it will fit his needs and makes no attempt to find out has no business bellyaching. That's why I deal with MISOSYS - I know Roy Soltoff will do what is right I am grateful that MISOSYS (and LSI) is there (here?) to provide support for the TRS-80's. I have never failed to receive a response to any question I have posted on this board. Usually both Roy Soltoff and LSI support respond [not to mention our resident SYSOP and guru, Joe KyleDiPietropaola. -ed]

I'll end this by saying THANKS ROY and LSI support. I don't usually leave a 'thanks' after I have a question answered (not wanting to take up valuable board space or your time) but it is always implied and heartfelt.

Now that you have your laser printer all set up and running fine, maybe you could re-release the MARK IV collection using the printer to print the documentation from your files for each order. Expensive. I'll pay it. I don't care that Pro-PaDS, Z-Shell, Z-Cat, and FM's date functions won't work w/ 6.3. The MARK III collection on my model I spoiled me. PLEASE! I'll grovel, beg, give you my first born male child, I'll pay full price for everything from now on (no more ZBASIC trade-ins) - anything! (Did it work?)

I never would have thought that my little comment about EnhComp would generate so much traffic. Understanding why floating-point produces 'errors' is not difficult. Programming w/ FP is. With my Mod I, I use to stick the numbers in the right place and call a ROM routine (Thanks to INSIDE LEVEL II). If you write an article for the TMQ how about a FP library for EDAS.

Fm MISOSYS, Inc: We can include a QRC but not a tabset - they're too big.

The products in the "Collections" were dropped because too few were sold and the docs ran out. If the products sold better, we would get the docs re-printed. Getting that done is not a problem; finding a reason to do it is.

No, we are reserving the laser for new stuff. Sorry. And with our third due in June, our house is full of feet. You can hang on to your first-born <grin>.

CTL255 Filter

Fm Lynn Sherman: Just got my copy of the Winter 1988 TMQ. Since the weatherman was talking about possible snow this weekend, it seems to be a mistake. Surely this must be the summer issue! All kidding aside, its a great looking issue.

Laser printers (LED array printers) sure do produce nice output. I'll have to look into the difference - never heard of LED array printers. Glad I procrastinated enough on my article to make the first laser issue. I notice that I managed to screw up the proportional spacing on one line by using the word "telecommunicative." The article used up even more space than I thought it would. But I didn't want to just give a program listing, especially since I wasn't very satisfied with my documentation of the code. I've always been from the "San Georgio" school of programming. I aim for a nice layered lasagna, but end up with vermicelli that's been run through a paper shredder. Chio for now.

Fm Shane Dawalt: Lynn, your article wasn't that bad!! In fact, I really enjoyed reading it. I don't have PRO-WAM, and I haven't done any serious programming on my M4 since the end of November, but I do recall the many messages that floated by here about CTL255/FLT. Interesting seeing the finished product. (Now, isn't assembly language programming fun?? <grin>)

CTL-255 & Foreign keyboards

Fm Lynn Sherman: Roy, I just received a letter from a TMQ subscriber in Belgium. He was reporting a problem with the CTL255 filter that I had not even dreamed of. He has two model 4's, a standard desk model 4, and a 4P with a French keyboard (didn't even know there was such a thing!). The standard 4 works fine with CTL255 but the 4P with the different keyboard does not. He did not describe the problem beyond this, but the assumption is that the keyboard matrix differs. He thought that he might be able to fix things if I could supply him with the program I wrote to show what bit is set in the matrix by each key press. I will be mailing him an assembly listing of the program. Do you think there would be any value in publishing the program in TMQ? Are there any

other unusual model 4 keyboards that you know of? I didn't include the listing in the original article because I thought it was of limited interest/application. But if there are multiple matrices, perhaps others might need it as well.

Fm LDOS Support: Lynn, There are a range of foreign machines, including French, German and other versions. Some have what are referred to as "dead" keys for generating accented characters. The key by itself does nothing, but the next key struck gets the added accent. Les did the TRSDOS 6 drivers for the foreign keyboards, and so might have some additional information.

Fm MISOSYS, Inc: Lynn, there are actually three international versions of Model 4s: French, Belgium, and German (I believe). But his matrix should be similar. I do know that they have a few extra keys, and the actual position of a few keys may be different. I think I may even have a matrix layout for the international machines from my development days on 6.0. I'll take a look. Now see how you have gained fame and notoriety? -Roy

Fm Lynn Sherman: Roy, Thanks for the information about Model 4 keyboard versions. I'm a little dubious with respect to the keyboard matrix being the problem, unless there are also special, international versions of Superscripsit, or unless SS does a hardware check and uses a different keyboard scheme for different keyboards. When time permits, I can probably check the second possibility but I'll have to ask around with respect to the first. I'd be very interested in that matrix layout for international machines. I remember that there were a few unused bits in the matrix with my standard keyboard. I suppose the extra keys on an international keyboard would use some of those. Its pretty tough to get at the root of the problem when you don't have access to the offending hardware/software combination and the problem is described as "it doesn't work". N'est ce pas?

MC -> PASCAL: No!

Fm Rex A. Basham: Roy, I got the TMQ I.ii & I.iii on Friday. Thanks a bunch and I hope you don't think me too much of a pain for the multiple phone calls. Receiving them was the only decent thing that happened on an otherwise lousy day. I managed to tear an expensive sweater, had an all around bad day at work, wrecked my beater vehicle on the wag home, and to put icing on the cake, I got a ticket for 'failure to yield the right of way' in the process. I must say I've seen better days! The $200.00 deductible was destined for double sided drives but I suppose they're back burner material for now.

Until I read your discussion of postage costs and whose questions get answered it had never dawned on me to include the SASE when I write. Sorry, a mistake I'll try to avoid in the future. Couple of questions for you or anybody else out there. On page 104 of TMQ I.iii there is a mention of Kenmore Computer Technologies Z-TIME1 clock (Fm: Jerry Wagers). Never heard of them. Where do they advertise and/or do you have their mailing address?

Next, you mentioned you would consider a PASCAL compiler for publication. Were you serious and were you referring to a Model 4 version? It couldn't be that hard to overhaul MC for the syntax required by PASCAL. Could it? If the compiler was semi-smart, the remainder of the MC package should be useable as is. After all, if the compiler is kicking Out p-code, a p-code converter should be a fairly simple piece of software used to generate Z80 source code. If I remember anything from my college days, the parser and lexigraphic scanner are the hard parts and you already have those in MC.

RE: TMQ I.iii page 77; I'll look into converting RAMDISK to using the @EXMEM SVC. Should be an interesting piece of code. Take care and JB says hello to Brenda.

Fm MISOSYS, Inc: Rex, Yes, that Friday did seem rather strenuous for you; Fridays aren't supposed to be that way.

I really don't know where Kenmore is. If you want to query Jerry Wagers about that ZTIME1 clock, you can write to him at 8405 56th Street, West; Tacoma, WA 98467.

As far as trying to re-write the MC source code into something suitable for a PASCAL compiler, the work involved would be excessive - perhaps not anywhere near as much as starting from scratch, but nevertheless, too involved to be considered. Remember, our program development efforts for the TRS-80 have been eliminated. And the set of files encompassing the MC package are so massive when you consider the libraries, that it is a great burden just managing the set of files. We just wouldn't consider such a venture - especially in light of the poor opportunities for selling a package. I would recommend to someone that they would be better off buying a cheap PC clone and getting Borland's turbo Pascal. But thanks for the thoughts.

Fm Michael Rogers: Roy, please find enclosed 1 TMQ coupon for "Lair of the Dragon". Yes I am actually buying a game for the Model 4, perhaps the last to be published?

Congratulations on the new look TMQ, I hope it continues publication and coverage relevant to the Models I/III/4 for some time yet. How gratifying to find my PRO-WAM contributions published in the last 2 issues.

Now that everybody (LSI 'creatures' included) know that it is 1988, that they should be using LS-DOS 6.3, LDOS 5.3, the date patched version of LDOS 5.1.4 and which software is Logical Systems' and which is MISOSYS', would you be interested in any of the following for publication in TMQ?:

(i) A step by step description of how to customize LS-DOS 6.3 for the beginner or intermediate user - what the options are and how to install them. People I know with Model 4's regard SYSGEN as a mystery beyond them and have not read the

DOS manual in sufficient depth to discover what the options are, the reasons for using them and how to go about getting them to work.

(ii) A humorous article on why the TRS world has just about gone kaput, with some encouragement for newcomers and those who have not realized what good computers they have.

(iii) A 'Down Under' column to especially cater for your Australian readers with programs specific to Australia. For starters I could supply some BASIC programs which would have Particular interest for readers in Australia, which may at the same time have programming ideas of interest to all readers. There could also be room for news of the TRS-80 scene here - perhaps not much as Tandy Australia sold its last Model 4 nearly 2 years ago. But people could advise of sources of software and other services still available in the different parts of Australia - I have some information of that nature. I would be happy to act as 'editor' of such a column. After the initial effort by me, interested contributors in Australia could send material to me for submission to Tmq.

(iv) A review of "Lair Of the Dragon" - it might be favorable!

(V) A review of 'LeScript 1.80 - with Spelling' Checker'. I know this is a product from another company, technically in competition with yours but their continued existence and support can only benefit all still involved with the I/III/4 line. YOU have probably heard that there have been problems with the spelling checker, but each copy they send me gets closer! 'LeScript' is a very good word processor even without the spelling checker and when the bugs are finally squished in its dictionary handling algorithm it will be an even better program. A favorable review in TMQ may help keep the TRS-80 community chugging along a little longer. Who knows 'ANITEK' might buy advertising?

Please let me know if any of these ideas appeal and I will get to work. No payment is required - perhaps software vouchers or similar.

P.S. (Yes! In these days of word processing: a 'p.s.'!) How do you pronounce MISOSYS? Is it MY-SO-SIS? Is it MICE-O-SIS? Is it MY-SOS-IS?

P.P.S. Still can't get PRO-WAM to REMOVE when running Anitek's 'SuperDrive'!

Fm MISOSYS, Inc: Michael, Your five suggestions are quite suitable for TMQ input. Let me elaborate by point.

(1) I do get enough feedback from TMQ readers to ascertain that a lot of what is in TMQ is over their heads technically; down to earth articles will be useful. Thus, a step-by-step article discussing configuring a Model 4 DOS is right on the money.

(2) Nothing is worse than a magazine which is 100% serious. Unless we can laugh a little, our faces will become rigid. Humor is always needed. That's why I'm big on "letters to the editor" since they invariably give us all something to laugh at.

(3) Our Australian readership represents the fourth largest, albeit not necessarily a large readership (1 is US, 2 is Canada, 3 is Great Britain, 4 is Australia). I've always had an affinity with Australia and hope to visit there one day. I believe this feeling stems from my youth when I saw a movie entitled *On the Waterfront*. The story was about a nuclear holocaust where the only remaining nation was Australia due to its isolation from the other land masses. I think American readers would benefit from a 'Down Under' article. After all, we have benefited from Olivia Newton John, haven't we?

(4) Since the demise of 80 Micro, we feel a need to turn TMQ into more of a general magazine. We are starting the acceptance of advertising in the next issue (II.iv) and believe that reviews are acceptable material. I would not be adverse to a review of *Lair of the Dragon* even if it were not favorable. As long as it was **accurate**, I would accept it.

(5) Since we are taking advertising from other sources, why not reviews. I have always referred to TMQ coverage as *the MISOSYS product line and related machine environments.* Give me an accurate review of LeScript and permit me to provide Anitek with a response and I would feel it suitable for TMQ.

There you have it. You have done your homework in coming up with suitable material. But we do expect to start paying for articles.

Incidentally, MISOSYS is pronounced "MY-SEW-SIS"; it is an acronym for Microcomputer Software Systems. I didn't want it to be MY-SAW-SIS.

Fm Michael Rogers: Roy, regarding my suggestions for TMQ input, I doubt I will be able to get anything submitted for the next TMQ as I am in the middle of relocating 3,470 km (2150 miles) from Australia's smallest mainland state, Victoria (just over twice the area of virginia) to its largest, Western Australia (nearly 4 times the area of Texas). I was amused by the references to Australia and its isolation in your letter. Perth, the city I will be moving to is reputedly the world's most isolated capital city, 2750 km (1708 miles) from the nearest other state capital, Adelaide.

So much for the geography lesson, now for the movie lesson! "*On the Waterfront*" is Marlon Brando saying to Rod Steiger, "*I coulda been a contender!*". The film you were thinking about concerning Australia being the last surviving country after a nuclear war was, "*On the Beach*". I should know, I was in it! The film was partly made in the seaside town near where I lived with my parents and I appear as a child extra in the background of a scene between Ava Gardner and Gregory Peck. Incidently, Ava Gardner is reported to have said that

Melbourne (capital of Victoria) was a good place to make a film about the end of the world. Australia's tourist potential appears to have improved somewhat since then! Should you and your family ever make it up here (why must we be North orientated all the time?) and you need any assistance please feel welcome to get in touch. You would've had an expert guide to Eastern Australia in me, but now that I've gone West, I suppose I will be a bit of a stranger myself for a while. However being an explorer by nature I expect to use my new opportunity to see more of this great country.

TMQ input for the third issue this year will be forth coming.

Fm MISOSYS, Inc: Michael, I stand corrected. Funny how 20 odd years of time cause one to forget all of the facts - unless one has a strong reason to remember, such as you did. Perhaps its time to run on down to the video rental store and take out a copy of "*On the Beach*" for the VCR. I'll await the "coke bottle scene".

By the way, your letter is just some more proof to the saying that the world is closer than we think. Remember the saying, "It's a small world"?

On the Family Update

Fm Elmar von Muralt: Dear Brenda and Roy, I'm sorry I didn't respond earlier to your request for an extra US$ 10.00 for S&H. The Receptionist at work had already written out a cheque for $10.00 but it was of the Australian variety, which wouldn't be of all that much use to you, so your bill ended up in the box with my personal mail.

My wife Madeleine, our two sons Alexander 8 and Nathanael 6-1/2 and I were away on holiday in Switzerland, where we migrated from in '73. The objective was, to show the boys at least once in their childhood a "White Christmas", - too bad we picked the warmest winter since 1946 or so -. Even without the white stuff in the low altitudes near Zurich we had a ball staying and feasting with the rellies. Early in December we visited Madeleine's brother who lives in the french part of Switzerland in a village about 850m above sea level, where they had just a little snow. Some farmers had just put some liquid cow manure onto their fields. As we got out of the car Alex sniffed the air and said: "Never thought that snow smelled that funny!"

So there you go: you've got my vote in favour of the "Family Update" in TMQ.

I very much appreciate the TMQ, actually I prefer the plastic binders of the photocopied back issues, as this allows laying flat or even folding back of the pages, but I suppose it's not economical to do it on the regular issue. I also like the frequent use of subtitles, it helps when I'm looking for an item that I've seen before. Suggestion: How about an index (once a year) according to various topics raised by either articles, letters to the editor or chat from BBS - please.

With the demise of coverage of TRS-80 related items in 80 Micro it sure would be good if TMQ could carry some ad's from suppliers who haven't followed Tandys' lead in abandoning the TRS-80 family. Might I suggest that it would even be to their advantage, as your publication seems to be much more tightly targeted towards TRS-80 users and space in TMQ might not be as expensive as in 80 Micro.

In TMQ I.iv, page 44, 2nd paragraph (The LSI LS-DOS column) the writer (the writer doesn't identify him/herself - is it W. Schroeder as identified in TMQ II/ii?) states: "Tandy will probably have mailed out update notices to registered TRSDOS 6.2 owners by now ...". Well, they have not, or maybe they just missed me? Maybe either LSI and/or MISOSYS could follow up on this in order to make themselves known to more "missed out" III and Model 4 owners to sell LS-DOS 6.3, LDOS 5.3 and all your other goodies.

I really like the way LSI have enhanced Basic - great stuff for editing, especially if one wants to make things look a bit better arranged. One thing has puzzled me so far but I haven't spent any time looking for solutions yet: On boot up the system asks for date and time and requires 24 hour style entry. Why is the timestamp in 12 hour mode, requiring one more character position? Can I change that with a System command or by any other means?. Please pass this on to LSI.

Just in case you're interested what I've got here it is: Tandy 4P Gate Array with a Star DP515 132 Col. Dot Matrix Printer which is alright for printing tables and the like but for letters it looks too stingy. That's why I prefer to use Madeleine's NBI System 3000 wordprocessor which is hooked to a Ricoh RP 1600 daisywheel printer. It's a pity I can't use the Ricoh with my 4P as the Ricoh "talks" 12 data bits in parallel or something else weird (Can't find the info right now). Unfortunately the NBI Level G hasn't got a spelling correction else I really like it. Is there anything in the MS-DOS world (if I ever get there) which is like it or even better. Come to think of it, do you know of any comprehensive comparison of wordprocessor features and how to get hold of it?

In my last letter dated 12.Nov.1987 I asked about some info about some "Fax"- software for a 4P. Have you saved the answer for the TMQ II,iii? Looking forward to hearing from you.

Fm MISOSYS, Inc: Elmar, As far as Tandy notifying you of patches, most folks from the Australian computer clubs have told me that Tandy has left Australia. They were supposed to take care of the international upgrades for TRSDOS 6.2, according to Bill at LSI.

From your previous correspondence, I don't know of any FAX boards for Model 4 usage; there are quite a handful for the MS-DOS world. That's why I didn't respond to your query.

Announcing New Products

Tentative release of DED86 Version 2.0

By the time you receive your TMQ, MISOSYS will most likely have ready for shipping , a new version of DED86. Rich has incorporated most of the features our users have asked for and many which they haven't even thought of.

Its an easy job to alter the attributes of a file with the new <ALT-A> command. This provides the prompt:

```
Alter attributes(a----w): <A>rchive, <S>ystem, <H>idden, <R>eadonly, <ENTER> ?
```

Since directory record decoding is a lot easier done by the program, a command to display the statistics of an entry - be it active or erased, gives the following kind of information:

```
REMM.SYS: a----w, 11/15/87 21:22:06, first cluster=2891, size=15643
```

To make the job of file recovery a lot easier, DED86 now sports an option to position to a free cluster. This sub-command allows you to point to the next, previous, first, or last free cluster. Since the directory record statistics tells you the first cluster of a file (even if the file was erased, that still tells you what the first cluster was), and the file size, you can easily determine if the file was contained in a single cluster or more than one. If the file was longer than a single cluster, just make use of the <ALT-F> command to examine only the free clusters for fragments of the file. Once you have found all of the pieces, you don't even have to bother updating the FAT, just KEEP the clusters you have identified in the DED86 "keep" list and write out a new file.

In case you want more information about your disk, the added <ALT-D> command gives you a display of disk statitics such as the following example from my AST machine:

Bytes per sector	512	Reserved sectors	1
Sectors per track	17	Hidden sectors	17
Bytes per track	8,704	Reserved/hidden bytes	9,216
Heads	5	Number of FATs	2
Sectors per cylinder	85	Sectors per FAT	64
Bytes per cylinder	43,520	Bytes in FATs	65,536
Cylinders	771	Root directory entries	512
Total sectors	65,518	Sectors per cluster	4
Total bytes	33,545,216	Bytes per cluster	2,048
Media descriptor	F8	OEM name	MSDOS3.3
Directories	27	Files	745

Usage	Clusters	Sectors	Bytes
In directories	29	140	71,680
In files	8,951	35,804	18,331,648
Allocated	8,980	35,944	18,403,328
Free	7,325	29,300	15,001,600
In error	0	0	0
Locked	35	140	71,680
Total	16,341	65,389	33,479,168

Another significant new feature incorporated into DED86 is a set of ten auxiliary buffers. Each can be used to save a sector or memory image for later recall: <ALT-S> saves to an aux buffer while <ALT-R> recalls from an aux buffer. Using this facility, you can essentially copy sectors from one location to another.

It's been compiled using a new memory model available in Microsoft C 5.0. In spite of all these added features, DED86 is smaller and faster. The price remains the same: $59.95 + $5 S&H. Send back your version 1 disk for a free upgrade.

LDOS and Model I/III Information

Model I LDOS

A Model I 5.3???

Fm T. J. Hodges, *3711 S. Hereford Lane, Philadelphia, PA 19110*:

Roy, I want to let you know that I am pleased with LDOS 5.3. The additional features and the date extension have allowed me to make further use of an old computer for a couple more years. I am also a faithful reader of *THE MISOSYS QUARTERLY* and intend to remain so until you drop support of the Z80 machines.

I am running LDOS 5.3 on my Model I. That's right, my MODEL I. Everything is working fine, including HELP, BASIC, the updated library commands and directory date/time stamps.

When you required 1000 pre-paid orders as a condition of development I knew I would have to do something on my own. This was confirmed by your responses to Model I owners in TMQ I.iii.

I purchased a copy of LDOS 5.3 for the Model III as a starting point. Initially, directory dating was my only concern, but after reading the update documentation it seemed worthwhile to devote time to convert the entire package. Now, after a little work LDOS 5.3 functions as documented. The magnitude of the patchwork is about the same as that needed to produce Profile One + HD in volume II of the LDOS Quarterly.

MISOSYS products were all the tools needed to analyze the programs and make the conversion. DSMBLR and FED were the primary software tools and the LDOS Operating System manual with the quarterlies provided the vital information.

Due to hardware differences in the Model I it was simpler to use the SYSO/SYS file from LDOS 5.1.4 with a few small additions as the resident code. All other files provided with version 5.3 work with the appropriate patches.

Most changes are simply address changes in the model III files. The device setup routines require the greatest work since the Model I device tables are different from those of the Model III.

I've retained a few routines from version 5.1.4. I always invoke the SYSTEM (UPDATE) option because I often work through midnight.

In addition, the RDUBL routine is mandatory to drive the Radio Shack doubler. With the old SYS0/SYS file these routines work as originally written. I chose the easy way to install these modules by using a configuration file created under LDOS 5.1.4. Additional features can be SYSGENed under LDOS 5.3 if needed.

Two library commands require fairly large patches that increase the size of the code so I chose to put them in a separate file. The UPDATE and BLINK options require a high memory driver in the model I. The model III handles such things in ROM. An alternative is to patch the library files with (Lx, X'nnnn'=.....).

On the way to adapting the new system to my machine some questions arose. Perhaps you could enlighten me on these subjects.

The first is the function of the model I machine flag (MFLAG$). It is mentioned in the manual's technical section but not explained, (at least its not explained in the 5.1.3R version with 5.1.4 updates). I haven't found references to this in Volume II of the LDOS quarterlies or any issue of TMQ either. Therefore I haven't used it in the conversion. Could you help with this?

I used an area of SYSO/SYS (5.1.4) between X'46E8' and X'46FF' to add Flags and new entry points. Is this an unused memory area?

Another problem popped up while using DSMBLR. The addresses became fouled when I ran the program on BACKUP (v5.3.0). Labels were not affected. The program is assembled out of order and the disassembler got confused. It the first time that I've seen this problem. It is repeatable.

I'm using Profile One +, SuperSCRIPSIT, Visicalc, several language compilers and BASIC programs that I've used for years. Initially, an occasional error condition would arise that

wasn't handled properly but all were easily traced and remedied.

Fm MISOSYS, Inc: Kudos to your efforts. I thought I would provide your address herein so that other Model I folks could communicate with you. Perhaps you'll share your efforts. Now let mwe take your questions in order.

The MFLAG byte contains a code to designate machine specifics. In the Model I, it was set to indicate the presence of the Lobo LX-80 expansion interface. You've probably examined the code to sufficient depth to note that the SYS0 initialization routine sets bit-0 of the MFLAG$ if the Model I is booting off of the LX-80 E/I. The DOS uses this flag in a few spots to permit certain features present only on the LX-80.

The area in low memory which is unused is between X'46EA' and X'46FF'; the floppy disk driver ends at X'46E9'.

Now the reason why DSMBLR messes up labeling on a disassembly of the BACKUP utility is because BACKUP is a very strangely assembled module. BACKUP consists of a two-section root module and two distinct runtime modules neither of which executes at the origin to which it was assembled. Since the functions of backup by files and mirror image backup are mutually exclusive, BACKUP can save memory space by overwriting pieces of itself. Here's how that works. The root module contains common I/O code as well as code to handle the initial dialogue. the other two modules handle mirror-image and by file backups. Once the dialogue session is complete, BACKUP has determined the mode to be mirror-image or by file. It then moves the needed module to origin at the initial address of the dialogue code. By doing that, BACKUP minimizes the space it takes up in memory during execution and can hold a larger number of cylinders during the mirror-image backup as well as a larger number of file sectors during a by-file backup. Since pieces of the CMD file are assembled at load origins different from the exectution origin (use is made of LORG in EDAS), it effectively confuses the disassembler. One way to have countered that would have been to use CMDFILE version 2. You could have stripped out the various sub-modules of BACKUP, then relocated the LORG'd modules to their execution address, then disassembled those pieces. How's that for a solution?

Now why didn't we go through all of the motions to put together a 5.3 version? There's many parts to that answer - not the least of which is the quantity of folks out there willing to buy such an upgrade. The biggest reason corresponding to that *opportunity of sale* was the additional work we would have had to do to bring the disk interface up to current state of the art. Our fundamental design criteria for a Model I release was direct booting in single or double density using either the Percom-type or Radio Shack doubler board. That would have required major surgery to SYS0 in order to fit in a double density driver as well as major surgery to the BOOT cylinder to install the necessary code. BACKUP would also have needed major surgery to painlessly reformat track 0 for single density so that it could boot on a Model I. That level of work compared to the opportunity of sale precluded us from making that investment of time. Too many other things needed our attention. Of course, now that you have accomplished your goals, others should be able to capitalize on your success. Good luck.

RSHARD

More on RSHARD

Fm Jon Roberts: Roy, Just a quick question on your RSHARD package. Does your formatter use two physical tracks per cylinder as does the radio shack one does? this pertains to the radio shack 15 meg primary.

Fm MISOSYS, Inc: That depends. Combining two physical tracks for one logical track is needed if you want more than 202 cylinders on a single partition. So there is no clear answer to your question. RSHARD permits you to partition by cylinder, by head, or by both. You can have 153-cylinder partitions, if you so desire.

Model III LDOS

LDOS LOMEM

Fm Michael Kushner: Does anyone know what to patch to enable setting "lower" high memory when using LDOS 5.3.0? I know this is a strange request, but I have a strange need. At the moment, the limit would seem to be x'73EE'. I would like unlimited low memory HIGH$ setting, but would settle for around x'2FFF'. Alternately, what to patch on LDOS 5.1.4 to allow dumping below x'5500'? I can set memory as low as 2B3B, but cannot dump. I can dump anything on 5.3.0, but can't set HIGH$ as low as I would like. Any solutions?

Fm Joe Kyle-DiPietropaolo: Michael, Don't forget that both MEMORY and DUMP are library commands, and thus must load into memory from disk to execute. This will clobber at least some of the regions you are referring to. I would guess that not all of the protected regions would be trashed, however.

If you have FED II, you can examine the modules for these commands directly to find the tests. If noody else has already suggested a solution, I can try to peek on disk when I have a chance.

LDOS 5.3 FREE

Fm Richard VanHouten: When all 8 drive slots are active, FREE with no parameters pauses after drive :7 and re-displays the info for drive :7 calling it drive ":/" after you hit a key. I tracked down the cause of this to a CALL at address 53C3H that should be a JP (CD versus C3.)

Fm MISOSYS, Inc: Richard, you're exactly right. Actually, it was originally designed to be a CALL to fall through to the following routine. But some later code revision in the routine being called called that following routine. Changing the CALL to a JP sets things right. The line counter was being decremented twice for each single line displayed. After the 8th drive, the command paused. This resulted in a loop to the next drive which was beyond the limit. Since there is not enough patches to make up a Patch Corner in this issue, here's the fix:

```
PATCH SYS7/SYS.SYSTEM
(D0C,59=C3:F0C,59=CD)
```

JCL Inconvenience???

Fm Louis Self: I end my JCL files for my Model 4 with "//EXIT" to pass control to a selected called program after the JCL file is finished its excecution. However LDOS 5.3 needs to access the JCL file one more time at a very inconvenient time: the first time the called program accesses the disk. In some cases this is later after the disk with the JCL file has been replaced by data disks. In that case the garbage is input into the system and the result is unpredictable. If the disk with the JCL file has been removed then the computer goes into never-never land with the cursor continuing to blink. This can be unverving or disasterous when persons other than myself are using the computer.

Is there anyway to fix this? Thanks for your help.

Fm Jim Beard: Louis, It is always a NONO to move the disk with the /JCL or .BAT file while the JCL or batch job is executing, under ANY DOS.

Fm Joe Kyle-DiPietropaolo: Actually Louis, the final line from the JCL is only executed when/if the called program does an input or line input that would "see" JCL. If you can force one of those (i.e. your own program), that's one possibility.

Another possibility would be to use TYPEIN rather than JCL for some of these cases, as TYPEIN does all of its disk I/O before it starts to feed things to the system. Make that TYPEIN for Model 3 mode is on Utility#1.

Fm Louis Self: It seems to me that the DOS should be written so that when it sees the //EXIT it reads the final line of the JCL and closes everything up. What is gained by delaying until the called program does an input? At best this is an inconvenience with BASIC programs and an impossibility when calling most commercial machine language programs.

You mentioned using TYPEIN. That would defeat the whole idea of the JCL which is to avoid having error prone novices typing in commands.

I love LDOS and LS-DOS but it seems to me this is a bug. Is it?

Fm Les Mikesell: Louis, JCL does exit as soon as it sees the //exit line. However, it does not read anything until a line-input request is made. If the program invoked uses only @key or @kbd (basic INKEY$), JCL has no reason to look at the next line of its file. If the program is your own, put a lineinput at the beginning. TYPEIN will also supply input to the single-key requests and can be used from within a JCL. I don't see how that would defeat your purpose.

Fm Joe Kyle-DiPietropaolo: Louis, What you describe is exactly what is done. Unfortunately, JCL **can't** read the last line from the JCL until the previous line is finished, which it won't be until either the program so executed on that line either releases control back to DOS, or does a line input.

Quiz time: Why?

BUG? Hardly, as it has been a fundamental design feature present since LDOS 5.0.0 (before if you count VTOS).

You misunderstand my reference to TYPEIN. TYPEIN is a utility program that is designed to read information from a disk file and feed it back via the keyboard. Sounds like JCL, you say? Somewhat, he says, with a major difference or two: (1) TYPEIN will feed to INKEY style requests in addition to INPUT requests. (2) TYPEIN does all its disk I/O up front, not at request time.

TYPEIN for the Model 4 mode is on the LS-Utility Disk, and TYPEIN for the Model 3 mode was on Utility Disk I.

LDOS Clock display

Fm Brad Stiles: Re., the Hardware Interface Kit. I run most of my home applications on a Model 4 (or 4P) in the Model III mode, The interface Kit has allowed me to operate in a fashion very much, like I operate the office 4P in the Model 4 mode, with one exception. The clock in the Model 4 mode, after installing LS-DOS 6.3's SYSTEM (HERTZ5) command is very accurate. The Clock in the Model III mode (LDOS 5.3, SYSTEM (FAST)), runs at 1/2 of the correct rate. NoW that LDOS is free of the ROM dependency (due to SYS2MEM), is it possible that you will develop a patch that will fix the LDOS clock for 50 Hz operation on a Model 4 (SYSTEM (FAST))?

I have tried to fix this clock problem with TIME50/ASM, from LSI Journal, Vol 2, No. 5 (Jan 84). The only relocatable assembler that I have is PRO-MRAS. The code assembled fine. When transferred to the LDOS system disk, it seems to install fine, but has absolutely no affect on the clock. I posyulate that I have two problems: 1) TIME50/ASM is intended to fix a Model III clock, LDOS (SYSTEM (SLOW)); 2) the compilation of LDOS source code with a LS-DOS compiler does not work. If possible, please comment on my postulations.

Fm MISOSYS, Inc: Brad, I believe that Tandy revised the MODELA/III file for the LS-DOS 6.3 release to take care of the clock in fast mode. But LDOS takes care of a timer adjustment; thus, you have the situation of two separate adjustments which then cause the clock to run half fast (don't say that too fast).

Here's at least one solution. I dug into the code in the MODELA/III file. The clock timing routine starts at address X'3556' and is as follows:

```
LD    HL,4216H     ;Point to counter
DEC   (HL)         ;Decrement
RET   NZ           ;Ignore until 0
LD    (HL),30      ;Restuff max count
CALL  3197H ;Change to 60 if "fast"
INC   HL
.
```

The model 4 normally operates at a 30HZ interrupt rate in slow speed. That's the reason for the countdown from 30 (30 interrupts per second). When the Model 4 is switched to its fast speed, interrupts occur at 60HZ. Thus, the revision in this release of MODELA/III fixes up the clock speed problem for Model III mode. Since LDOS already fixed it up internal to its interrupt code, you can restore proper time by patching the CALL 3197H instruction to be ineffective. The easiest way is to change the byte at X'355D' to a 11H. That will change the CALL instruction into a "LD DE" instruction which is acceptable there.

This should correct your problem. If you are using a different version of the MODEL? file, look around that address range for the *correct* location.

Faulty LDOS 5.3 Diskette?

Fm a customer; *April 6, 1988*: I purchased LDOS 5.3 from you on your invoice number 88-10766, dated 12.17.87.

I backed up several of my working diskettes and started having trouble using QFB and BACkUP to copy my diskettes. Recently I worked over two days trying to get a group of programs on two diskettes.

I finally ran Kim Watts's PCHECK on your diskette after it told me I had errors in all of my copies. It gave me the following: HIT byte 27H invalid or extraneous. When trying to use BACKUP, it aborts the backup at: loading cylinder 33 - Parity error during read.

In desperation I ran Kims PFIX on the disk. That solved the HIT error, but it still gives me the same error when using BACKUP. I would appreciate it very much if you would send me a good copy of the system.

Fm MISOSYS, Inc: There is no HIT error in LDOS. Kim's PFIX diagnosis stems from his assumption that an entry in the hash index table for a reserved SYSTEM file is an error - it is not. You should not have removed the write protection from your master disk and written to it. We have no assurance that any problem occurring with that disk is yours, ours, or otherwise.

I checked your disk on a number of machines and disk drives and detected no parity error; the disk was entirely readable. Your purchase was made on December 17th and your first complaint of trouble was April 6th; that's in excess of three months. If you would like us to refresh the diskette, there is a service charge of $10 + $2S&H. That is our normal fee for out-of-warranty diskette repair.

DOS doesn't BOOT?

Fm Donald Strumpf: Roy, I am a registered owner of several MISOSYS products and on occasion I have called you with a question or so. You have always been helpful and friendly. Recently, I had a problem which may be trivial to your more sophisticated users but I believe it is an interesting one.

I have a Model III that has seen many years of heavy use in my law office It is mainly used for word processing. Your new DOS was installed in January (nothing like planning ahead). My LDOS 5.3 was up and running almost immediately and it ran beautifully for a couple of weeks. Sometime in late January, without any warning, it refused to boot up. I tried all the tricks that I knew - cold boots, warm boots, other copies of LDOS, etc. Nothing worked.

My Tandy dealer suggested that I try TRSDOS, which I did. It worked! He then told me that it was a software problem, not a hardware problem like I suspected. Not being satisfied, I called you (on the pay line of course, not the 800 line). You suggested that it sounded like a hardware problem but you couldn't pinpoint it for me.

I was confused as to what to do next. So, I got on the phone again and started calling my buddies in the local Tandy user group (Philadelphia based PACS). The eventual breakthrough came when it was suggested that I try my factory original of LDOS 5.3. It worked too! This was the clue to the solution of the problem. I had two DOS's that were set up for single-sided drives that worked (TRSDOS and the original LDOS 5.3 disk). My working LDOS 5.3 disk had been set up by me for

double-sided drives. It sounded like one of the heads on my double-sided drive zero had given up. The drive was an old full-height Tandon that has needed alignment off and on. Therefore, I decided the time had come to replace it with a modern half-height drive.

Fortunately for me, my user group had a scheduled meeting shortly after this revelation. After the meeting, I bought a Toshiba drive under the supervision of a guru from the user group. Armed with detailed instructions from several user group members, I decided to install the drive myself. It wasn't as easy as I had been told, but I completed the job nevertheless. My vintage Model III was plugged in and the working copy (double-sided version) of LDOS 5.3 was put into the Toshiba drive. Success! It booted up perfectly, I and my secretary have been using the revitalized Model III for several weeks now and it has operated flawlessly.

I know my story does not contain the deductive reasoning present in a Sherlock Holmes' tale, but the inferential chase was still fun. Let me know if you think it would be helpful for your readers to hear the details of how a klutz finally installed a disk drive into his computer.

Does DIR/SYS need a date?

Fm Mark P. Fishman: Roy, Having finally moved one of my TRS-80 Model I's to a convenient place, I have discovered a curious thing. I had occasion to try to read a file under LDOS 5.1.4 (patched with DATEFIX/JCL) which had been written with LDOS 5.3.0 on my Model 4. I can read the directory. I can read the file. I can even load the file into ALLWRITE and PSCRIPT (PowerSoft's patch to SCRIPSIT).

But PSCRIPT and ALLWRITE refuse to list the directory of the disk! Apparently, while LDOS is oerfectly happy to use the new directory structure, programs that open DIR/SYS as a file are running afoul of the change to where the access password used to be.

In as much as DIR/SYS (as a file) is maintained without a date anyway, are there any dangers in using Super utility or other methods to change the 0000 back to 9642 (bytes +18 and +19 of the DIR/SYS directory entry)?2 I cannot think of any, but I thought I'd ask the designer.

I understand that I should not write to a 5.3 disk with a 5.1 system, and usually use 5.1-formatted disks for exchange. Once in a while, though, being able to read a 5.3 disk directly is a real convenience, and being able to check the directory inside a word-Proceseor even more so. Note that I do not need to convert the disk (UNDATE as in TMQ) nor make a new copy of the original file, merely refer to an old document. This modification to the DIR/SYS entry seems harmless and a brief experiment indicates that it appears to solve my problem. Perhaps you'd address this in TMQ?

Fm MISOSYS, Inc: Mark, DIR/SYS doesn't need a date entry since it is never updated - at least not by the DOS. You bring up something we actually overlooked. If DIR/SYS doesn't have a word value in the date field that would look like a blank password under 5.1.4, then a 5.1.4 system would not be able to open DIR/SYS as a file without the password of .SYSTEM since 5.1.4 would reference the old access password field.

You could also patch the FORMAT command of 5.3 to create the DIR/SYS record with that value. Then you would not have to worry about modifying disks which you wanted to look at via a 5.1.4 program which opened DIR/SYS as a file.

Special Character Mode

Fm David J. Kelton: Roy, When I wrote to you on January 23, I asked if there wasn't someone who knew the address in memory where the space compression/special character switch was located. In TMQ I.iv Pg 32, Paul Bradshaw noted that printing a CHR$(28);CHR$(194)* -- CHR$(2l) for the Model III -- and then checking the value of POS(0) would tell the user whether space compression or special characters were in effect. I didn't like this approach because (1) the cursor had to be at the left margin when the print statements were issued, (2) it appeared awkward, and (3) it doesn't work in Assembler.

I have discovered the address in the Model III where the switch is located. In reviewing the Model III "Operation and BASIC Language Reference Manual" supplied with my unit, I came across the ROM routine $INITIO (X'0069') whose stated purpose was to initialize all the I/O drivers to their default conditions. Using DSMBLR, I found that this routine loaded the contents of adresses X'36BF'-X'36D7' into addresses X'4015'- X'402D'. I verified the defaults against a partial listing of tbe addresses given in the manual. Then I wrote a simple BASIC program to peek at these addresses; toggle the space compression/special character switch; and peek at the addresses again. When done, only one address changed value. Address X'4024' was set to 0 for space compression and set to 1 for special characters.

I don't have a Model IV which is what Paul Bradshaw and Pete Betz were using, but perhaps this same technique might locate the appropriate addresses for them. For me, I'm finally happy because: (1) IF PEEK 16420 = 0 THEN space compression ELSE special characters; and (2) IF (INP(255) AND 8) = 8 THEN special characters ELSE alternate characters. This allows me to neatly and cleanly set everything the way I want it without any extraneous output to the screen. No April foolin'.

Why KI/DVR for SPOOL?

Fm MISOSYS, Inc: The *Adelaide Micro User News* posed a query in the April 1988 issue. Let me reiterate the dialog.

"NA - You need KI/DVR in and running for the SPOOLER to work. The spooler is an INBUILT COMMAND!! What's a keyboard command doing mucking around with spooling?? GB - TRUE - The documentation states the KI/DVR is required to set up the <CLEAR> key as a control key for the SPOOLER, but how it is used I don't know. EdG - That makes three of us. Perhaps one of our LDOS experts could let us know why you need a CONTROL key with the LDOS SPOOLER. Strike one for Nick!"

Here's the reason for that requirement. First of all, it has nothing to do with the use of CLEAR as a CONTROL key. The LDOS spooler is, in fact, two processes in one. It is a spooler; but it also is a despooler.

The function of despooling is an interrupt task. You cannot do disk I/O during an interrupt task for a number of reasons. First, interrupts are always disabled immediately on receipt of an interrupt and must remain disabled until the interrupting process completes. Second, a disk data transfer takes longer than the time between interrupts. Third, the floppy disk driver will enable interrupts on completion. Thus, an attempt to do floppy disk I/O within an interrupt process will usually cause a machine lockup; the system will not support interrupting an interrupt task.

Once we recognize that some of the print output being spooled does, indeed, overflow to disk, how then can that disk overflow be despooled to the printer if the despooling process is an interrupt task? The answer is that the entire despooling process is not an interrupt task.

In order to provide a *safe* means of getting that information off of disk during the despooling process, the DOS hooks into the keyboard driver. Invariably, programs issue keyboard calls. When they do, that keyboard hook is strobed which then allows the spooler to check on the need to take some spooled-to-disk data and read it into memory. If the keyboard is not polled by a program, then despooling could stop. The hook is part of KI/DVR; if that driver is not installed, the keyboard hook doesn't get activated and despooling from disk cannot occur.

DOS errors from BASIC

Fm Jim Dague: I have located a new little problem in 5.3 and maybe you can help we out with it. When in LBASIC in 5.1.4 the program used a CMD"COPY and the name of the file:0 :1" etc. It continued to send the files until an ERR 122 (disk full) was sent. It all worked very well. When the same program is run in 5.3 it appears that "Disk full" no longer is considered an error and no number is sent to memory location 409AH and the program continues to send files to the disk even while it will display a Disk Full message on the CRT. When finished a check of the directory shows the names of the files have been added to the directory but of course, you have all 0's because the files couldn't be copied. My question is how does one get the copy program to stop in 5.3 when the disk is full when in BASIC? Note: I want the program to continue the copy process when a new formated disk is placed in that drive, which it did in 5.1.4. Thanks for any help you may be able to provide.

Fm Roy Soltoff: That problem has been reported already. The problem may not necessarily be solvable. It is based on the fact that 5.3 introduced a @CMNDR vector to invoke library commands and return. In this facility, all library commands were coded to always return via a RET statement even if an error exit was to be taken. Under an error case, the error code is put into the HL register pair. Now this is defeated if JCL is in effect and an error is directed to @ABORT. BASIC, on the other hand, traps errors coming back from the CMD executor by trapping the @ABORT vector. Now if I "corrected" BASIC to consider an error return if the HL register was non-zero, any program you invoke from BASIC which does not adhere to the new convention and happens to have HL non-zero on exit will then be interpreted as returning from an error condition. So its a real dilemma as to exactly how to proceed. For the one individual who reported the condition (he also was using COPY until a disk full error), I gave a one-byte patch to the COPY command to override the error return in COPY and always drop through to @ABORT (as if JCL was in effect). I don't have the location handy in SYS6, but the RET Z (C8) at 548EH in COPY (SYS6/SYS) could be NOP'd out. As he rushes to another machine to find the exact spot... It's D20,EC in SYS6/SYS. Change it to a 00 and COPY will then always go to @ABORT on an error condition.

LS-DOS and Model 4 Information

TRSDOS 6.2 & Date/Time

Model 4 '88 dates

Fm Roy Watson: It appears that my TRSDOS 6.2.1 will not accept the 1988 date. Why?

Also, I am using a hard disk, I seem to be unable to turn on the "TIME" prompt at IPL time. I do boot from floppy and then the system resides on the H.D. .. What is the trick in turning on the Time?

Fm LSI(Virgil): Roy, The date range of TRSDOS 6 is 1980 through 1987, as documented in the TRSDOS 6 manual under the DATE command. There is a release of TRSDOS 6 that extends the date range up through 1999. This release of the operating system is called LS-DOS 6.3 and is available from us, LSI, for $39.95. [Effective June 1st, 1988, LS-DOS 6.3 will be available and supported by MISOSYS, Inc.] It is easy to install LS-DOS 6.3 on your application program system disks by following the directions provided with LS-DOS 6.3. Please be sure to make backup copies of your disks before you update them, just in case.

LS-DOS 6.3 is NOT an update to Superscripsit, General Ledger, Accounts Payable or any other application program. Most application programs take advantage of the date feature of the operating system, so updating the operating system on your application program's system disks to LS-DOS 6.3 will, in effect, also extend their date range also. LS-DOS 6.3 does not directly change your application programs or your data files.

The "trick" to turn the time prompt on or off when using a hard disk is that the hard disk boot floppy must be the "system" disk, temporarily. Assuming that you boot floppy drive is normally referred to as logical drive :4, boot up your hard disk as you normally would and then do a

```
SYSTEM (SYSTEM=4)<enter>
SYSTEM (TIME=NO)<enter>
```

and then re-boot.

SYSTEM(TIME)

Fm Fred M Gray: Joe, Using the following command,

```
TIME (CLOCK=Y)
```

the time is displayed at the TOP RIGHT of the screen. Is there any way to change the screen position PERMANENTLY so it will print, say, at the bottom center? (Reason exists.)

Or, could you identify the SYS module, record and relative byte, which might be zapped to give this effect?

Fm Joe Kyle-DiPietropaolo: You can Fred, but you don't want to. The display area under the clock isn't saved and restored before scrolling, so you'll end up with "clock droppings" all over your display. If you still want to try it, I'd be happy to look up the location for you. This would only be useful for an application that does not scroll the line that the clock ends up on. Scrolling and redrawing doesn't count either.

Fm Gary Phillips: I like that, Joe: "clock droppings". Sort of like those little dead cursors you can leave all over the screen of a model 3 if you do a lot of PRINT@...; without turning the cursor off first! Probably you could relocate the clock then by doing an interactive "patch" from the application... AND making sure to handle scrolling correctly so as to pick up the doo-doo from the display.

Fm Joe Kyle-DiPietropaolo: Yeah Gary, isn't that great? Unfortunately, I can't take full credit for it. Somebody else coined a form of the term when referring to the incompatibilities between certain versions of applications software and some manually manipulated interface control element device driver code. That's "mouse droppings".

Fm Gary Phillips: Well Joe, it's certainly a logical extension. (More logical than using a rodent has ever seemed to be from my point of view.)

Fm Joe Kyle-DiPietropaolo: Ok, Fred, as per your request here's the scam. The address in question (the location for the clock display) is loaded during the boot process as part of BOOT/SYS (lowcore). Look for the hex sequence 21 45 F8, the last two bytes are the address in standard Z80 lsb msb

format. This is the physical RAM address, as the start of video ram is brought in at x'F800'.

When in RAM, this code will land somewhere near X'0780', and will move from version to version of the OS.

Note that the clock data string is eight bytes, so don't attempt to exceed the end of the video ram (start no higher than x'FF78') to lop off the seconds or anything kinky like that. You'll mess up the type ahead buffer at x'FF80'.

Model 4 JCL

//STOP

Fm Debbie Turner: I am using a JCL under LSDOS 6.3 to enter a menu in BASIC. The JCL is as follows:

```
//IF BOOT
SYSTEM (DRIVE=2,DRIVER="MEMDISK")
D
D
Y
//END
BASIC (M=65303)
RUN "DMENU/BAS"
//QUIT
```

When I get into the menu, I enter a selection, then at the next prompt takes me to Job done back to LSDOS. Anyone know how to stay in BASIC until the menu is exited?

Fm MISOSYS, Inc: It appears that you have not terminated your JCL with "//STOP". If you want to remain in BASIC at the conclusion of a JCL sequence, terminate the JCL with "//STOP" and not "//EXIT" or "//QUIT".

DOS Interfacing

IEP in LSDOS 6.3

Fm Richard Watkins: I am working on an IEP program that will sort of crash proof a bbs by loading it back up in the event of a fatal crash that returns to DOS. I did this by copying my program into SYS13/SYS after copying the existing SYS13/SYS to a backup file. After exiting my IEP program the original is restored. I controlled the Eflag (IY+4) using a SVC call from 'C' and set it to 1 for my program to run and then set to zero when I exit and restore the original IEP (or ECI, I usually use a shell program). I have noticed that several different shell programs and other assorted ECI programs manipulate the flag by making it other non zero values. What I would like to know is if there is any way to force my 'C' compiler to load the restored ECI program. I am using PRO-MC and I use the cmdi() function to load the actual bbs but I need to load the program as if the user pressed '*' to load it since different ECI's do different things. Is this possible or not worth the effort?

Fm Joe Kyle-DiPietropaolo: Richard, If you check the 6.2 doc, you will see that a number of the bits (four I think) of EFLAG$ are passed along to the ECI when invoked, the ECI can use these bits for internal control functions.

You might try saving the EFLAG$ value before you change it and the ECI, then restore it along with the ECI when you are through. Forcing a reload of SYS1 from disk upon exit from your 'C' program might then do it.

Did you get your problem with peek and poke worked out?

Fm Richard Watkins: Joe, Where can I save the ECI flag? I don't know that much about the memory locations that are used. IS there any place in memory I can safely poke one byte and then read it ack without fear of it being manipulated by the bbs program?

I got my 'poke' worked out. I was confused about pointers, for one thing, but I think I understand them now. I used the memset() function to poke the byte in memory at IY+4. It seems to work, although for some reason the bbs program I have isn't working very well.

Fm Joe Kyle-DiPietropaolo: Richard, You could try UFLAG$. That's the "user flag". It is possible, but unlikely, that the BBS program would try to use that.

Fm Richard Watkins: Joe, I'll try that. I suspect that I have a wild pointer in my program or some other bug. I used the system() call to route the printer to a disk file in the eci program and it isn't working. Nothing is getting stored in the file and every once in a while the computer locks up leading me to believe that I have something messed up, as usual.

Fm Joe Kyle-DiPietropaolo: Richard, That might be a problem - doesn't ROUTEing to a disk file require allocation of a memory block from HIGH$?

Fm Richard Watkins: Joe, Could be. If I route the device outside of the program. Evidently it does something strange when I try it for 'C'.

Fm Shane Dawalt: Yes, Richard. When a ROUTE is performed, the system allocates a 256+ byte buffer in high memory. This is used to buffer input/output to/from the file.

Fm Roy Soltoff: Of course it will, Richard. The way around that is to redirect the printer device to a disk file at the DOS level then reset *PR. Thereafter, when in C you can redirect *PR again via a system() function because the DOS will reuse the FCB and file buffer it had previously established.

Now I just had a second thought, the system() function in C may not be able to find anything in high memory above itself. That was not added to the function. The problem is that a save-and-relocate facility such as the system() function must add the high memory module header so that @GTMOD can skip past it to find anything else in high memory. I put that memory header facility into LDOS BASIC's CMD processor but it was not put into MC's. Perhaps I'll look into that one day.

What I previously said above about re-using the file buffer would have worked if C's system() function inserted a module header.

Mod 4 Special characters

Fm Larry Prall: How can I determine whether the special characters are active on a Mod 4? I know I can find out whether the primary or alternate set are in use by reading port &FE, but there must be someway to make certain that the special characters as opposed to the space-compression characters are in use. Actually, my immediate problem is to control the state rather than determine it; i.e. if I can turn the special characters on and off absolutely rather than through a toggle I don't have to be able to determine their previous state. Would like to know, though, if there is a way I can do both.

Fm Larry Prall: [Guess what, folks,] I found the answer in "*The Programmers Guide*". It kept getting past me before for some reason, so I did some experimenting. I went into TASMON, dumped low memory, sent &H15 out port &HEC and dumped memory again. Did this a few times and determined that bit 3 of the byte at &H0B94 was getting set when special characters were active. I went back to the Guide to look for the address, and there I found all the information I was looking for. Sigh...

Fm Adam Rubin: Larry, One method is to move the cursor to the start of a blank line, write a special/space-compression character to the screen, and see how far the cursor has moved afterwards (one space for a special character, or several for space compression).

Fm Larry Prall: Adam, Appreciate your help, but my problem was to have my software detect the status. I have since found out how to get the status and to change it. The CORRECT way to do it, without assuming anything about versions, etc, is to use the @GTMOD service to locate the device driver ($DO), then add four to the address returned in DE to get the first byte of the data area. Bit three controls the status of the special characters vice space-compression. Again, thanks for taking the time to respond.

Model 4 Hardware

Tape recorder access

Fm Charles Halliman: Does anybody know if a cassette recorder can be attached, and controlled by the model 4? Please give me some feedback if you can.

Fm Richard Buchman: Charles, I don't know about the Model 4D but a regular Model 4 has a 5 pin DIN plug on the rear (next to the power cord) which you can plug in a cable (supplied by Radio Shack of course) to operate a cassette recorder for data tapes. Don't use mine much for anything except voice games though.

Fm Joe Kyle-DiPietropaolo: How much control, Charles? You get a relay closure for motor on/off via the cassette jack on the back, but that's about it.

Fm Mark Mueller: Joe, The Model 4 only supports the cassette port in model 3 mode, unless you use the casette to transfer filed from a Model 100 via the TAPE100 program. Otherwise, about the only thing you can do with it under model 4 mode is listen to the relay click.

Fm Fred M Gray: Mark, Another use for the CASMOTOR relay in the 4: you could use it to switch which outgoing telephone line you would like to use. I've put it to such use: when I'm working, I use a "prospecting program" in conjunction with a (Little Brother) database; when I select a record containing a telephone number, then I can automatically send it to the SMARTMODEM for the simple purpose of dialing the call. If my local area code is in the telephone # string, then nothing special happens. If it's local to ANOTHER phone line, which I happen to have, I can save long distance charges by switching in THAT local-call line. The relay and some hardware outboard does the trick.

Fm Mark Mueller: Now that's interesting, Fred We have two lines at work. Maybe I'll fiddle with that a bit.

Application Programs

PRODUCER Software

Fm Bjorn Thogersen: To keep track of wines in a winecellar, I am using the PRODUCER Data Management software, published by "Software of the Future", on a TRS-80 Model 4.

The publisher no longer responds to my inquiries, so I am wandering if by chance some of you would know enough about this program to help me with my problem, which is:

I have added records to the program to the point, where I am about to run out of space on one disk. I have "deleted" many records, but even deleted records take up as much memory as active ones. Deleted records can be overwritten by new ones, but that process is much more time consuming than the addition of totally new records.

So my question is: How can I get completely rid of obsolete records without having to start all over with a fresh disk and re-entering records? Would appreciate some thoughts on this matter.

Fm Joe Kyle-DiPietropaolo: Bjorn, You somehow need to do what is called "pack your files". That means moving the good records "forward" over the deletes, and then truncate the files to remove the excess space. Are you sure that this isn't an option somewhere in the package already? If not, you will need to know the exact format of all the files that are used to contain the database information.

Alcor Pascal + Profile4+

Fm Jerry L. Barker: Help! Here is the background. I am using a Model 4 128K system. I am writing a program using Alcor Pascal ver 2.00.00 to read files created with Profile4+ ver 1.00.07. The program works fine with RUNP. What I want to do is to call my program (readfile/cmd) from a user created RM/CMD, and then return to RM/CMD after completion of my program. Going to my program was easy to add, but I am lost as to how to get back to RM/CMD. My best guess is to use the SVC external procedure with Pascal, and to call @RUN SVC. What has me stumped is how to get the pointer to the FCB for RM/CMD, as described in the Technical Manual.

Another thing I am unsure about is if the RM/CMD remains open waiting for a return from programs. All Profile4+ programs return, but I don't know how.

One more problem that I am having is trying to create a CMD file. The program works great under RUNP when I allocate 4k to stack. What seems strange is when I end the program, it shows 0 amount of stack used. When I run it under LINKLOAD, I have to allocate a fraction of k. If I allocate 1k or more I get an error message that the stack is too large. When I create the CMD file using LINKLOAD, the program will not run through. I have tried allocating from 0 to 15k when building, but no good. The system always locks up in one form or another.

Hopefully, you are familiar with Alcor Pascal, or if not, perhaps you can refer me to someone who is. I know nothing of assembly language, so please make any suggestions simple, if possible. Thanks!

Fm Gary Phillips: Jerry, I may be able to give you some help, though I don't know anything about Profile4+ and have never used it. I am very familiar with Alcor Pascal for both model 4 and model 3, and have used them extensively. I don't know if there is anyone else here who uses Pascal much, so I'll volunteer.

It sounds as though your problems with LINKLOAD involve some confusion about how LINKLOAD works. Have you ever successfully built /CMD files with LINKLOAD and used them? I'm not quite sure what you mean about an error message saying the "stack is too large." LINKLOAD doesn't issue such a message. Do you mean that you get this message during execution of the /CMD file you created with LINKLOAD? Is there an error identification number? All programs created with LINKLOAD will require at least a minimal stack, you can't use a stack size of zero. The minimum stack size is the total space required by all variables in all routines that are active at the same time, plus 32 bytes for each file variable, plus 64 bytes for standard INPUT and OUTPUT. What happens if you let LINKLOAD default the stack size (just press enter when it prompts you for stack size)? Are you sure you are answering the stack size prompt correctly? (If specifying 1K, for instance, be sure the K is upper case.) The message about how much stack was used when a program terminates refers to dynamic stack allocation, and you may be doing nothing to use dynamic stack space (do you have subroutines in your program, or call system routines?)

You say your program works when you invoke it via RUNP, which tells me your code is basically OK. In my experience, anything that works with RUNP will work with a /CMD file built by LINKLOAD, though not necessarily the other way around. If the program works with RUNP but freezes the system when you build it with LINKLOAD, either you have a corrupted copy of LINKLOAD (try using BACKUP to get a fresh copy from your master disk) or have included a defective copy of one of the system libraries (SYSTEM/OBJ, STRINGS/OBJ, or RANDOM/OBJ). Try getting fresh copies of the libraries from your master disk as well.

I don't know what RM/CMD is. Is this part of PROFILE4+ or something you wrote yourself? You may not be able to call a Pascal program from another program and then return, because Alcor Pascal will overlay all of memory, erasing the calling program. You won't want to use the @RUN SVC because that is intended to run an external program and then return to the calling program rather than passing control permanently to the caller. However, if you can get back to your menu or whatever RM/CMD is by starting RM/CMD over at the DOS Ready level, then you <u>can</u> use the @CMNDI SVC to get there. @CMNDI requires a pointer in HL, not to an FCB, but rather to a command string like "RUN RM/CMD{CR}". Build this string in a PACKED ARRAY OF

CHAR variable. Get the value for HL by using the LOCATION function to obtain the address of the string. Then use the SVC procedure to pass control to @CMNDI, which does not return to your Pascal program at all.

A thought: if RM/CMD is invoking your Pascal program via the @RUN SVC, that could indeed crash the system. Do you freeze up right then or only after the Pascal starts executing?

Have I helped clear anything up? I check in here every day or two, so feel free to ask anything if you think I can help. If necessary, I could take a look at your Pascal code, or whatever.

Fm Jerry L. Barker: Gary, thank you, thank you, thank you! You hit it on the head! The @CMNDI worked perfectly, so I am up and running. Just for reference, RM/CMD is just a PROFILE4+ menu cmd, simple. And the @CMNDI works great to get back to RM/CMD.

Now the other thing I wanted to let you know is that LINKLOAD does have a STACK TOO LARGE message. I looked in the book and it is not in there, but it was on my screen. Here is what had happened. My program has an ARRAY OF RECORD.

When the number of records were set at 500, the program ran with RUNP at 4k, ran with LINKLOAD r command if the stack was set at .1 to .9k. Don't ask me why, it just did. If I set the stack at 1k or over, I would get the too large message. A CMD file would not run with any size stack. Now the actual file that was read was only about 100 records. Any time the program was ran successfully it terminated with a 0 stack used. Yes, I know that can't be done, but that is what it said. Here is what I found out. With the array set for 285 records, the program ran with a 23650 stack size under all conditions, including as a CMD file built with LINKLOAD. If the array is set for 300 records, the program would not run with anything, always giving a out of stack error message. But if the array was set to, say, 500, 1000, 1500, etc., I got what was described at the beginning. Just food for thought, and thanks a bunch! Now let me be so bold to ask another question.

Now that I am up and running, I am working on reducing the file size. As I said, the main object is an ARRAY OF RECORD, with the record containing several fields of PACKED ARRAY OF CHAR of different sizes. I have a sort routine to sort by different fields, according to user input. What is bugging me is that I had to include two complete sort routines to sort by the different fields. Do you know of a method to use a variable in place of the field name? This is what I want to do. BASE is the array, DES is one field, NOM is the other. Instead of sorting BASE[1].DES or BASE[1].NOM, how about VAR:='NOM', then sort BASE[1].VAR? I thought that it was a good idea, but I could find no way to get it to work. Any suggestions on how I can do this, or maybe another way to do the same thing? Glad to find out that I am not the only one in the world that likes programing with Pascal.

Fm Joe Kyle-DiPietropaolo: Jerry, I'm not that big on Pascal, but if that were 'C' I'd say that it sounds like the array is being generated as an automatic, and allocated from the stack at run-time. If the array is not defined at the very top, outside the scope of any function, you might try moving it there.

Again, I'm no Pascal maven, but in 'C' this would be a relatively simple exercise with pointers. You'd use a pointer to the base of the key field, then index the pointer by the size of the array element.

Fm Gary Phillips: OK Jerry, I think I understand the stack problem. RUNP was not checking you for a stack overflow. If you have nothing but static variables, including that large array, you might get away with it. LINKLOAD builds things in memory a little differently, and checks to see whether you can actually use your stack without smacking into the heap or the high memory pointer. That's where it was catching you. I strongly recommend that you replace your ARRAY OF RECORD with a linked list using pointers, and the NEW() and DISPOSE() commands. That way you are not preallocating any huge amount of storage that may not actually get used, and you can check as you go along to see whether you have run out of HEAP space. If you find linked lists too confusing to work with (they're easy once you play with them a bit) then alternatively you could allocated a static array that contains only pointers to the desired RECORD type. Then, as you need to fill an array element, use NEW to create the actual space off the heap, and let the array element be a pointer to the record out there in dynamic memory. I use Pascal to work with binary tree structures using linked lists and recursion. It's fast and requires far less code than the same job in Basic or Fortran. Glad I was able to help you out with @CMNDI, and will make a note in my manual about the undocumented error message from LINKLOAD. [Anyone else here who uses Pascal? Or are Jerry and I the only ones left? Actually I use Pascal on a VAX as well, to do everything that the C programmers do.]

Fm Gary Phillips: Jerry, There are several ways to work out your sort problem. Passing a variable name is not going to work in any normal Pascal implementation because after the program is compiled, the variable names don't mean anything. Joe's suggestion of using a pointer variable is a good one, but probably not the easiest approach in Pascal.

The first idea that comes to mind (I sometimes think in assembly language, which is where this algorithm comes from, though you don't need to know any assembly language to use it) is to use a free union. If as an example we take this record:

```
type
sample = record
   case boolean of
      true : (fld1 : packed array [1..4] of char;
      fld2 : packed array [1..20] of char;
      fld3 : packed array [1..6] of char;);
      false: (flda : packed array [1..30] of char;)
   end;
```

(If you aren't familiar with variant records, you'll want to study them in the manual AND in a good textbook.)

This rather odd-looking definition lets you reference sample.fld2 normally, but in your sort routine you would instead reference fld2 by using sample.flda[5] and pass the parameters in as the offset of the desired field (in this case, 5) and the length of the field (in this case 20). All of these values can be defined via type or const to make the actual source code easy to write. Most flexible sort routines written in any language will use a similar technique (i.e., reference the sort key by length and displacement). Does that help?

Alcor Pascal does indeed support variant records and free unions fully. C programmers also call this thing a union, by the way, though they call our "record" a "struct".

Fm Jerry L. Barker: Well Gary, I am going to look into the linked list and pointers as you suggested, but let me ask a few questions first. To begin with, the program works great, the files are already around 200 records, and will probably stay around there. I hope anyway. My sort routine will sort by a 35 char field, 200 records, in about 4 seconds, which is fine. Is using pointers just good programming practice, or will there be a tangible advantage, such as being able to handle more records or smaller program size? My last dealings with pointers was in college, which was farther back than I care to admit, so I must bone up some. You mentioned that by using pointers that I could tell when getting close to capacity by checking the heap. With my current program, I added a check to notify the user when within 5% of filling the array. Due to using the @CMNDI SVC call, I do not get a STACK & HEAP used message, so I would have a hard time checking it. Of course, there are ways to have the program check it as it is running, but I haven't used them.

My next experiment will be to build pointers in memory for the records on a disk file. For a simple screen run of the files would be simple, but sorting will be something else! But by using this, if I ever get rich and can afford a hard drive, the files could be combined into one large one, instead of several small ones like it is now.

Fm Gary Phillips: Jerry, When you create a large array with a VAR statement in Pascal, you are allocating that much memory permanently to the array for the duration of that routine and all routines invoked beneath it. In an application such as the one you describe, this can be very wasteful of limited memory resources and place constraints on the amount of executable code that can reside in memory at the same time. By using dynamic variables (pointers with the NEW and DISPOSE functions) you avoid the static assignment of all that space and instead guarantee that the program will only use as much memory as it actually requires. This is more flexible and could probably be characterized as "good programming practice" as you put it. The linked list is one way of creating an array structure in dynamic memory space (the heap), but is by no means the only way of doing it. Alcor Pascal has two functions for checking the amount of free memory available at any time during the program. I am at work right now and the manuals are at home, but I believe the functions are called MEMAVAIL and HEAPSIZE. Look at the SYSTEM section of the manual. One tells you the total number of bytes of free memory, and the other tells how big the heap is right now (or something like that). You can use these to dynamically monitor the size of the stack and heap, rather than having to "eyeball" that status message at the end of the program. This is much the same as your present check to see whether the array is within 5% of being filled.

Sorting a disk file of fixed length records by using an algorithm for sorting a memory array is not difficult. Read through the file, copying only the key field and record number into your linked list. Sort the linked list. Open a new output file, and read the original file in random sequence following the record numbers in the sorted list, copying the records to the output. Close both files, delete the old one, and voila!

DOS Speed

Is DOS 6 slow? Other DOSs?

Fm Pete Granzeau: Joe, Just so you don't get a big head over all the compliments, I have been conversing on another commercial system with a Model 4 owner who refuses to use TRSDOS or to buy LS-DOS 6.3. He uses CP/M exclusively. Claims that loading overlays for everything the DOS does slows it down too much. I am too technically unsophisticated to argue the points he makes. You wouldn't like to call Delphi and argue with the guy, would you?

Fm Gary Phillips: Shane, I take it you've never actually used CP/M. Compared to LS-DOS/LDOS it is extremely clumsy and limited. In order to achieve the functionality of the LS-DOS library commands, you need a whole diskfull of dedicated utility programs. Using these costs just as much time in loading programs as if you were loading DOS overlays. Even MS-DOS has to constantly reload COMMAND.COM, it seems. The primary reason for the LS-DOS overlays is limited memory. If you stick the overlays in a RAMDISK you eliminate that problem and it runs FAST.

Also, as Joe pointed out, the program loader in CP/M is extremely stupid. It only knows how to load object code origined at 0100H. If you want to origin your code anywhere else, you have to provide your own relocation routine, load at 0100H, and move your own code elsewhere.

Fm Shane Dawalt: Ok Gary, I see your point on the speed issue. I agree that the overlays do load quite fast from the RAMDISK, but it's all my 64K upper bank can do to hold the entire system. Besides, I normally use RAMDISK as a data drive for applications that use the disk almost constantly so that rules out placing LSDOS into the upper banks. It is nice, when I can do it.

Fm Joe Kyle-DiPietropaolo: Pete, I don't have a Delphi (or GENIE or The Source) subscription, so I don't think that I'll pop over and do much myself.

Fm Jeff Brenton: Pete, Just point out to him that, except for those 6 built-in commands in CP/M 3 (DIR, ERASE, RENAME, DIRSYS, TYPE and USER), **EVERYTHING** is an overlay. The main advantage of CP/M, though, is that all of the DOS functions, such as open, close, extend, read and write, are always resident. TRS-DOS derivatives use overlays to maximize TPA. And, the overlays are laid out in such a way as to minimize delay, since, if you need the overlay to extend the space allocated to a file, you need to access the disk anyway, which means the overhead of loading SYS8 won't be missed to much.

By minimizing what the DOS has in it, you can make for a very small, fast system. That's what CP/M is; it is little more than a program loader. LDOS and TRSDOS/LS-DOS (which is actually much closer to CP/M in size, because the ROM code isn't there) have a lot more incorporated into them.

If CP/M does everything that person wants it to do, more power to him! I could never put up with the restrictions of not maintaining the actual file size in the directory, lack of relocation information in the executable file format, and about a dozen other things that make CP/M the primitive system it is.

The LSI Column by Bill Schroeder

Important Announcement

Please read completely

For those who wish to know the current Version of LS-DOS 6.3 is still LEVEL - L+ - with file dates of July 1, 1987. There have been NO patches, changes or even a bug report in the last several months, so all is well.

Some of you may have LEVEL-K if you created yours through patching and some may have a LEVEL-L without the + designation. If so you should install the following two patches. Don't be alarmed if the fail to install, that only means you don't need them.

```
SPOOLER/FIX - 11/01/87 - For Level L SYS8/SYS.LSIDOS
PATCH #017

This patch is related to Patch #010

PATCH SYS8/SYS.LSIDOS (D21,E4=40;F21,E4=60)
PATCH BOOT.SYS.LSIDOS (D02,1F=2B;F02,1F=20)
```

Patch #010 was in level J but was left out of level L. It was put back in at Level L+, so it would be a good idea to verify that patch #010 is in place in your system, here it is.

```
PATCH BOOT/SYS.LSIDOS (D06,2C=40;F06,2C=60)
```

I should also remind everyone that you may still send in you disk for an update for just $5 and not have to worry about patches, but now you will send that disk to MISOSYS..... READ ON.

Now for a very important statement: LSI will be leaving the TRS-80 market place on JUNE 30th 1988. No, LSI is not going out of business. We will be directing our attentions to other things and MISOSYS will be selling and servicing LS-DOS 6.3 so you need not worry about support.

As of June 1st (1988) you MUST direct all LS-DOS 6.3 sales and support inquiries directly to MISOSYS only.

You can still contact LSI directly if you feel you must by writing to LSI c/o MISOSYS, or to MISOSYS, Attention: Bill. I will try to answer all letters.

LS-DOS 6.3 sales and support as of JUNE 1st:

MISOSYS, Inc.
LS-DOS Department
P.O. Box 239
Sterling, VA 22170-0239

703-450-4181 (hours 9a-5p M-F ET)

MS-DOS Information

Future Products?

MISOSYS MS-DOS Utilities

Fm Kerry Wilson: I have recently purchased a 1400LT. I have several of your utilities for the Model 4 series and would like to get similar utilities for use on the 1400. Do you have a MS-DOS version of most of your utilities? (I can't find my copy of your catalog right now cause it's still boxed up from our little "flood.") Are they supplied on 3.5" diskettes?

Fm MISOSYS, Inc: We don't have an MS-DOS version of most of our TRS-80 utilities. We do have LB86, DED86, RATFOR-86, FM86, GreatBac86, Lair Of The Dragon, ED/ASM-86, LBMU86, Bedford Accounting. A few more are in the mill. For instance, we'll shortly have a second adventure from Dave Goben, Colosal Cave; a DiskDISK-86 is in the works; plus some other things which will remain unannounced.

If you need programs supplied on 3.5" media, you need to specify at the time an order is placed. We normally provide programs only on 5.25", although we can support 3.5".

Editor

Fm Jim Beard: Roy, Have you considered a small, fast language editor similar in concept to SAID, for MS-DOS?

Fm MISOSYS, Inc: We have. It's still in the thinking stage.

MS-DOS Hardware

286 Express Board

Fm Jim Beard: I just installed my new '286 Express. I got it on sale from a store for $299, even though they had to order it. It came in 5 working days and installed in 10 minutes; I had to move my Memory+ board to the middle slot. It booted in '286 cache mode without a hitch right off the HD. My PC Tools says that my processor speed is 3.10, up from 0.95, relative to an IBM PC. My AT at work reads only 2.60. Procomm still works, but I had to increase the delay between transmitted characters from 50 msec to 165 msec.

Boy is the box full. I do believe that my old Tandy 1000A has seen as much as it ever will.

3.5" floppy drives on MS-DOS

Fm Peter J. Fournier: I would like a bit of help with a small problem. At work I recently installed 1.44 Mbyte 3 1/2 inch disk drives as drive B in our pure IBM ATs. Using PC-DOS they work fine as 720K drives, but no matter what I do they won't do 1.44 Meg. (BIOS is dated Nov ?? 85, PC-DOS is 3.3) I bought MS-DOS 3.3 hoping to use DRIVPARM instead of DRIVER in my CONFIG.SYS file. Wouldn't you know, I try running it and at bootup I get the message "Unknown command in config.sys". BULL!!! The command is well documented in the MS-DOS manual, everyone in the lab has looked at, read it, entered it, "Nortoned" it, and walked away scratching their heads. I don't mind not getting 1.44 Meg on my drive but I do mind not being able to use a documented command. Does anyone have any ideas?

Fm Joe Kyle-DiPietropaolo: Messy, Peter. The DRIVPARM CONFIG.SYS parameter was present only in 3.2 of both MS-DOS and PC-DOS, and was removed in 3.3 despite the documentation to the contrary. The doc is in error as per confirmation with Microsoft.

If you have the absolute most recent version of the AT BIOS ROMs (I don't know what the date is), I've heard that setting the floppy drive to an undocumented Type 4 in SETUP (assuming that you have a version of SETUP that lets you do this, or can otherwise monkey the CMOS) will run the 1.44 meg drives just fine.

Note that this only applies to machines using the stock IBM AT HD controller. Folks with Western Digital and other vendor controllers may have lots more headaches.

When using DRIVER.SYS, remember that you must use the alternate drive letter (probably D: in your case) to use the DRIVER.SYS parameters for formatting. Also note that you

must use real HD 3-1/2 inch media to get reliable access (About $4 to $5 a crack).

Lastly, if the drive you installed has the media sensing switch installed, it can tell the differnece between media types.

Fm jeff brenton: Peter, I'm surprised the 3.5" drives even worked at all in the ATs. I just went through all of that, and it took an update to the BIOS to get it to recognize the drive.

However, you have yet another problem. Machines not manufactured for the 1.44M drives will NOT switch between formats automatically. It takes a hardware control not present in 99.99% of the controllers out there to do the switching - you have to either wire it for 720K or 1.44M, **or** install a switch to let you control the line manually.

Browse through the files in DL 1 of the IBM Hardware Forum (GO IBMHW), using the key "*3*". I think you will find at least a half-dozen files on the subject of getting 3.5" drives in general to work in ATs, and some mention of how to fix up 1.44M drives in particular.

Fm Daniel L. Srebnick: Peter, Drivparm does not exist in DOS 3.3. It was never a documented parameter in the IBM DOS, only in some MS-DOS 3.2 versions. The best thing to do is use driver.sys (check the documentation) and have the drive addressed as D:. Be sure to set the switches on your system board to indicate only 1 primary floppy drive is present. Then the first drive will be known as a and b, the hard drive as c, and the 1.44 meg drive as d.

Hard drives

Fm Theodore Masterton: I understand, from my recent acquisition of my 15 meg for my 4p, that a hard drive system consists of a controller, a "bubble" (or the hard drive itself, a power supply, and some software. I also have learned that the software must be written for the controller, and **may** either <1> allow parameters to specify the nature of the bubble such as Mr. Soltoff's drivers, or simply is married to a hard drive/controller combination, as Monte Micro's CP/M drivers.

My question is about MS-DOS. All I ever see advertised in the drive and the controller (no drivers). Are the drivers part of DOS? Does that mean that all hard drives and controllers must be engineered to the drivers in MS-DOS?

More practically, I have been reassuring myself that paying for the new bubble for my flea-market 15 megger was a good investment because I planned to use it in my AT clone that will be moving in early next year. Now I am wondering. I saw an ad for an 8 bit HD controller that said "uses 5, 10, and 20 meg drives". That suggests that the controller must "know" the bubble. Maybe my 15 megger will be worthless.

Also, I am coming to understand that HD controllers in the clone world can be either 8 bit ($60-$120) or 16 bit ($150+). I bet that the 16 bit controllers only work in the AT's? Is the economy of an 8 bit controller in an AT a foolish move?

Fm Hardin Brothers: Theodore, I can answer a few parts of your question:

The driver software for the MS-DOS hard drives is on the controller card itself, in a ROM chip. When a PC-type computer boots, it looks through memory for such ROM chips, does a quick checksum, and then hands control over to the ROM to install its own code, hook itself into the machine's BIOS, and initialize the hardware. Then the ROM returns control to the boot-up sequence in the computer's BIOS.

I'm not sure if an 8-bit controller will work in an AT-class machine, but a 16-bit controller certainly won't work in an XT-clone because there are only 8 data lines to the card socket. However, if you have a 286 machine, it would be foolish, in my opinion, to try to get an 8-bit controller to work. For many programs, especially database stuff, disk access speed is more important than processor speed in the program's overall performance. The few extra bucks will make a big difference in performance, and therefore in your appreciation of the machine. Most AT clones that I've looked at include the hard disk controller as standard equipment anyway, combined on a single card with the floppy controller.

Fm jeff brenton: One other note, Hardin - From what I've seen, the AT-type controllers do NOT have the full control software on board; that's in the BIOS already. What IS on the board is the drive type table.

What this suggests is that an 8-bit card will not work in an AT-type machine (and not all 80286/80386 machines ARE AT-type anymore, what with the accelerator boards out there) because the BIOS would not look for its on-board controller BIOS.

Fm Hardin Brothers: Jeff, I'm not sure an 8-bit controller would work in a 16-bit machine, but the BIOS would probably link itself into the system correctly. All standard BIOSes that I'm aware of look through memory starting at C000:0 for ROMs. Any ROM can be put at any 2K border from C000:0 through DFFF:0. If the ROM follows the rules (has the correct signature and correct checksum), control is given to the ROM to install itself, which usually means hooking into the necessary interrupt vectors. If a 16-bit machine failed to make appropriate interrupt calls for hard disk services, then a lot of products (large disk controllers, Bernoulli boxes, tap drives, CD ROMs, etc.) wouldn't work on that machine. Almost all of them install themselves at bootup in a similar fashion and grab some interrupts for their own use.

Fm Shane Dawalt: Hardin, You mean each ROM in a system has a sort of 'boot strap' code which installs itself into the system automatically? What book did you read which told about all this?

Fm Hardin Brothers: Well, Shane, each add-in ROM (the motherboard, BIOS ROMs bootstrap themselves rather nicely since they are addressed at the CPU's bootup CS:IP ((which is different on the 8088/86 and 80286/386))). But if the ROMs couldn't bootstrap themselves into the system, how would the computer know that an EGA card or a hard disk or an EMS card or ..???... existed? It's up to each device to log itself in. Some do it with device drivers (like most mice), some do it with ROMs, some with installation programs.

I can't remember when I first came across this -- I think it was a couple years ago. But I've seen it documented several places, and it does work. The process works like this:

The motherboard BIOS, after doing its own thing, starts scanning the memory space beginning at C000:0, looking every 2K bytes. If it finds a signature word of 55AAh, it begins to suspect it has found a ROM. Next, it looks for a length byte (# of paragraphs, if I remember correctly). Then it does a checksum on that memory for that length. If the processor's checksum matches the on-ROM checksum, the processor does a far call to start-of-ROM:3 so the device can install itself.

Therefore, if you want to install a custom ROM, or MS-DOS in ROM or anything like that, you merely need to set up the installation procedures correctly and away you go (well ... MS-DOS in ROM isn't quite so simple, since you have to get around the disk bootstrap routines in the BIOS which happen after the ROM installations).

Fm Ray Reyes: Shane, For all the information you could ever want to know, or not want to know, about MS-DOS, you might want to get the MS-DOS ENCYCLOPEDIA, published by Microsoft. It's about 1300 pages, and runs $135 in most bookstores. but if you're getting heavily into programming, it's a worthwhile investment.

By the way, Doug Hogarth, late of Powersoft and currently with Microsoft, is one of the contributors to the MS-DOS Encyclopedia.

Fm Shane Dawalt: Well I'll be darn, Hardin. That's slicker than ice cubes! I had no idea how the computer determined it had EGA or whatever. About the only thing I know is how to turn this beast on, turn it off, and insert/extract boards. I still haven't figured out what happens with something cooks up a big pot of bits on the CPU card. Hah! I'm still trying to decide what Desktop Publisher I'm gonna get (moderately priced Desktop Publisher you understand.) That means I'm fighting software installation yet ... don't go throwing hardware on me now. I'll need a lifeline. Hah!! But seriously, I do need to get hardware documentation on this thing. (This 'thing' is an AT compatable machine.)

Fm Joe Kyle-DiPietropaolo: Well, Jeff, actually the drive table is in the motherboard rom set on ATs. At least some eight bit drive systems will work in AT class machines, look at the Plus Systems HardCard40 as one example. I agree that it is false economy, and a full sixteen bit controller should be used.

Fm Joe Kyle-DiPietropaolo: Shane, Hardin's got the option ROM install procedure pretty well covered, I'll just add that it is discussed in IBM's Tech Ref Manuals for the AT and so forth.

Fm Shane Dawalt: Eh, Jow, Tech Ref Manuals? Question: Does that term apply to those "Inside the IBM PC" books which are in B Dalton? I believe they are by Peter Norton, wouldn't swear to it though. I guess I could get the yellow pages out and do some finger walking in the IBM section. (Need to burn off some of those calories you know.)

Fm Roy Soltoff: What you will find, Theodore, is that in a PC clone, the drivers for the HD are in a ROM chip on the controller. An AT BIOS usually has the HD drivers already.

Fm Theodore Masterton: Since I initiated this thread, I have been introduced to the setup program that comes with an AT (or is it with DOS? or DOS when it is supplied with an AT?) Anyway, I have come to learn that there are supposedly 47 types of drives recognized by SETUP. The Leading Edge at our agency has some sort of 40ms 40 meg seagate. A disk manager program with seagate's name in the introduction allows for several types of Seagate drives during installation. SETUP reports that our machine has a drive type of "32". The leading edge initialization manual lists some of the drive types from 1 to 47, according to heads, tracks, landing zone, and something else, but conspicuously omits a drive type from 31 thru 34. Curiouser and curiouser.

Anyway, on the pragmatic side, I see know drive type in the AT setup program for a 6 head 306 track drive such as I planned to remove from my Tandy 15 meg enclosure, so I am starting to doubt if it will ever find a home in my future AT clone. Maybe I will be able to sell the unit for the $275 that a new Seagate 225 would cost.

Fm Joe Kyle-DiPietropaolo: No, Shane, these are real honest to God from IBM Manuals, slipcases, ROM BIOS listings and schematics. Expensive, but useful for the ultra technoid. You can call 1-800-IBM-PCTB for a catalog, I think.

Fm Shane Dawalt: Wow, Joe, ROM BIOS listings!! Now that's something this technoid(?!) didn't expect. BTW, couple of weeks ago, I was hunting around for an 6800 book. Well, my Spring Quarter session just started and I had to get a 6800 book for a microprocessor course. Turns out, this book contains the 6800 and peripheral chips plus the 68000. Don't recall the name or who it's by. I believe the name is something like "THE 6800 AND 68000 MICROPROCESSORS". I did say "something like". The book is in a rather inaccessable spot right now so I can't look up the name.

Fm Joe Kyle-DiPietropaolo: Theodore, Drive types above number fifteen are particular to the BIOS/machine manufacturer, and another machine could easily have the drive type you need. Also, there are some software solutions, both as software and firmware.

You wouldn't want to put an ST225 in an AT - too slow to do the machine justice.

Fm Joe Kyle-DiPietropaolo: Theodore, Subsequent browsing indicates that the Tandon 15 Meg TM503 (which is what you probably have) is in as a Type 41 in at least one BIOS version.

Fm Theodore Masterton: Joe, Thanks for keeping my question in mind. Do you have a recommendation for a book that talks about the technical aspects of AT's? I need one that isn't exactly a fixit book, but more about the system conceptually.

Fm Joe Kyle-DiPietropaolo: Theodore, I know of nothing good that isn't very technical and/or very expensive. A good bet might be magazine articles in either PC Magazine or PC Tech Journal about the time the AT or the XT-286 came out, what, say late in 1984. There's an AT article in the Dec '84 issue of PC Tech Journal, but I don't have that issue.

Operating System

Disappearing cursor

Fm Shane Dawalt: Joe, I finally downloaded TAPCIS v5.1b. Have you noticed that the cursor disappears during data transmission while running automatic. I don't recall seeing any info on this disappearing cursor trick. It doesn't bother me, I'm just wondering if it's happening on your equipment too.

Fm Joe Kyle-DiPietropaolo: Shane, There are two methods to get rid of the cursor on an MS-DOS machine. First is to use the BIOS call for cursor size, the second is to move the cursor off of the screen. The former will not work on some machines, and the latter might not work with some screen "upgrades", like full-page displays. It was turned off on purpose, not to worry...

Fm Shane Dawalt: Ok, Joe. I was hoping it wasn't the software conflicting with my hardware.

Fm Ray Reyes: Shane, Many programs will shut down the cursor when doing "auto" runs.

Environment Variables

Fm Hardin Brothers: A short course in "Envrionment", "Path", and "Append":

If, at the DOS prompt, you type SET, you will see the environment: a group of strings in the form SOMETHING=SOMETHING.ELSE. Each string is held in memory in ASCIIZ (that is, null-terminated, C-like) form and each program gets a copy of the environment, not the original, from its parent, which is usually Command.Com. Turbo C has a couple of library functions that deal with the environment, including one (I think) which takes as its argument the left-hand "SOMETHING" and either returns a NULL or a pointer to SOMETHING ELSE. There is nothing magic about the environment -- it is just a way to pass information to programs. For example, Microsoft's languages use the environment to find the Include subdirectory, Library subdirectory, a TMP directory to use for temporary files during compilation, etc. The BRIEF editor (one of the best, IMO) makes very extensive use of the environment.

One of the environment variables, if you use it, is PATH. When you ask Command.Com to run a program, it first looks for a .COM, .EXE. or .BAT file that matches the command name in the current directory, then takes each directory in the path, in turn, looking for your program. For example, part of my path setting looks like this:

```
PATH=C:\;C:\BIN;D:\PROG\QB;F:\UTIL\TLIB
```

Many programs use the PATH setting to look for their data files. They simply get the PATH entry from the environment, then parse the string by looking for semi-colons or the terminating NULL, and try to open their data file in each directory. The usual technique works something like this:

(1) Try to open a data file for reading in the current directory. (2) If not found, get the PATH setting. (3) For each subdirectory in the PATH try to open the data file for reading.

When they find one file, they store the directory setting to that file and then use it to find all the others. Only after they have exhausted the PATH directories do the give an error message. That's the intelligent way to work, but not necessarily the only way.

APPEND is like PATH, but tells DOS, not programs, to look for data files in more than one place before returning an error to a program. You can think of it as sort of like TRSDOS or LDOS's ability to look on all active drives for a data file.

Argv[0] normally returns the complete path to the current program. You can experiment with it by simply writing a test program which prints its own argv[0]:

```
printf("My path is ==> |%s|\n",argv[o]);
```

you'll soon see how regular it is. To extract the path to a program's directory from it, you need only parse the string from right to left, looking for a "\" or, if that fails, a ":". Save that string and you have a direct route to your program (and subsidiary files).

One more thing: the word "path" can mean three different things depending on context: (1) the PATH setting in the environment; (2) the complete filespec for the file: drive, all subdirectories, and full file name (D:\PROG\TEST\T1\TEST.EXE); (3) the same as 2) but without the file name (D:\PROG\TEST\T1\).

And now, with Append, folks talk about the "Append path" or "Data path" which can also be like (1), (2), or (3) above.

All of which is probably more than you want to know but I hope it helps.

MS-DOS File Length

Fm Hardin Brothers to Shane Dawalt: The following program, compiled with MSC 5.0 under MS-DOS 3.20, produces a file, TEST.DAT, of length 0 on my computer:

```
#include <stdio.h>
FILE *fp;
void main(void)
{
   fp = fopen("TEST.DAT","wb");
   fclose(fp);
}
```

Fm MISOSYS, Inc: It should, nothing is being written. What is bad about MS-DOS is that you can't COPY a NULL file from one drive to another. That has caused us a few problems when you are preparing a set of files from a program and want to archive the set of initialized files. If it created some NULL-length files (by procedure similar to what you have portrayed), they won't get cop ed over to your "archive" diskette.

MS-DOS Utilities

MSDOS system RAMdisk

Fm Ken Kane: In MS-DOS 3.2 I have a RAMDISK utility from AST that sets up a RAM drive C:, no problem. Is it possible to make that the SYSTEM disk and have it accessed first? I do this all the time on my trusty B&W Model 4 with SYSTEM (SYSTEM=3). Any way to do this in MS-DOS?

Fm Hardin Brothers: Ken, Although "System Disk" isn't really an MS-DOS-type concept, here are a few things that will make the computer go to your RAM disk first:

- ☞ Keep C: as your default drive
- ☞ Make C:\ the first entry in your Path setting
- ☞ Copy Command.Com to C: and then SET COMSPEC=C:\COMMAND.COM

All of the above can be accomplished in your Autoexec.Bat file, in order to automate the process (I do the same thing with my laptop computer).

Fm Ken Kane: Beautiful, Hardin. Thanks for the clear and useful information. I have not tried RAM disking my AST 140, but thought it would be fun to try on my brother's Leading Edge system with 2 floppies.

Fm Joe Kyle-DiPietropaolo: Ken, MS-DOS always checks the current directory on the current drive first, so if you set that it should fine. Do note that MS-DOS keeps the OS pretty much completely in RAM, so no overlays are necessary. COMMAND.COM will be reloaded under certain conditions, and you can use COMSPEC to determine where it will be loaded from.

MS-DOS print spooling

Fm Gary Phillips: I avoid Mess-DOS if at all possible, but a friend is asking about print spooling. I understand that the PRINT command has a spooler, but it is somewhat deficient (incompatible with application programs, for instance). Is there a commercial or shareware software spooler, something like a TSR, that functions as well as the LDOS/LS-DOS spool command?

Fm Ray Reyes: Yes Gary, there's a package called PRINTQ which gives you mainframe spooling capability. All output is spooled to a disk file, and you can manipulate the order in which files will be printed, flag certain files for different forms (the program tells you when to load the new forms), cancel specific print jobs, and redirect print jobs to different printers.

Fm Shane Dawalt: I agree, Joe, but the ability to modify the print queue would be nice at time. Unless you get a fancy piece of hardware with a fancy piece of software to match, you aint gonna have that ability.

Fm Roy Soltoff: Gary, The PRINT spooler under MS-DOS is a DOS facility. Thus, I would expect that any program which outputs to the printer using BIOS calls will be bypassing the spooler. I use the ASTSPOOL facility which is part of the software which comes with the AST Premium/286. Works great. But the only negative with the version I currently am using is that it provides no command to totally dump the spooler buffer. I haven't yet upgraded the machine to MS-DOS 3.3 with the latest AST software utilities, so that version may have the spool dump command. I recently picked up a

Six Pak Premium multifunction board for my PC and the AST spooler with it did have a dump-the-buffer command.

Note that since the previous was written, I have upgraded the 286 to MS-DOS 3.3 along with the latest AST utilities. The AST SuperSpool facility now includes a parameter to dump the spool buffer. That's needed when you start a long print session and find you want to disregard the output. Dumping the spool buffer is better than resetting the computer.

Fm Gary Phillips: Roy, I'd forgotten about the AST spooler. I have a 6-pak board in the PC I use at work, but use printers on the VAX or IBM systems rather than a local printer so I'd never tried the spooler. My friend may even have an AST card in his machine (a Leading Edge D). What he really needs is something like the LDOS/LSDOS SPOOL command, but he uses WordStar and I'll bet the MSDOS WordStar bypasses the DOS on printing because it has its own internal spooler. A hardware buffer would probably suit Henrik, but his funds are limited.

Unfortunately, his willingness to grasp/handle technical details of either software or hardware is also limited. I'm afraid learning to use PRINT or SUPERSPL will put him off and he will buy a hardware buffer anyway.

Fm Roy Soltoff: Gary, That's exactly what the AST spooler is like. It doesn't spool to a disk file for later despooling. To coin a phrase from Randy Cook, it's a symbionic spooler. That means the spooling/despooling is concurrently in effect.

Fm Ray Reyes: Joe, Actually I've found Tandy's Spooler.sys to be resistant to TSR conflicts, but that's probably because it's loaded as a driver. You can start/stop/flush it, but there's no easy way to manipulate the queue. It's all in-memory, but can use extended/EMS memory if you've got one or the other; I have a 512K EMS print buffer installed.

I also have a hardware buffer -- a Quadram Microfazer with 128K -- but I recently picked up a Star NX1000 printer and the damn thing goes nuts if the Microfazer is installed. I can't figure it out and neither can the technicians at Star. So I have to fall back on SPOOLER.SYS. The microfazer is connected to my old Toshiba P1340 dinosaur.

Fm Joe Kyle-DiPietropaolo: Ray, I'd check the Microfazer outputs with a 'scope, especially the width and timing of the strobe pulse.

MS-DOS Applications

Desktop Publishing?

Fm Shane Dawalt: I'm still trying to decide what Desktop Publisher I'm gonna get (moderately priced Desktop Publisher you understand.)

Fm Joe Kyle-DiPietropaolo: Shane, Depends on what "moderately priced" means. Somehow I get the feeling that Ventura at under five hundred doesn't fall into that category...

Fm Shane Dawalt: Oh, Joe, I think $500 for a Desktop Publisher is a moderate price. I couldn't handle $1000 for something to generate graphics on though. That is a bit over inflated. Case in point (although it isn't a Desktop Publishing package), AutoCAD. $2800 (or so) for a program to draw with? I think not.

GEM's startup cost probably hits Ventura's price. I'm currently hounding the local DR supplier for pricing info on some GEM applications. (GEM DRAW +, GEM PAINT [maybe], GEM 1ST WORD +, GEM DESKTOP PUBLISHER.)

Fm Joe Kyle-DiPietropaolo: Shane, We just picked up another copy of the GEM Presentation Team here at work. Make sure that you get the latest GEM/3 version of the desktop and accessories, the improvements are significant. Several packages come with Bitstream's FontWare to generate fonts specifically for your output device in the desired resolution and point sizes.

Not bad, $315 for GEM/3, the Bitstream stuff, DRAW+, WordChart and Graph.

I'd skip DR's GEM Publisher, Ventura is a far better and more powerful package, and supports all GEM formats for graphics anyway.

Fm Shane Dawalt: I did query them about GEM/3, Joe. They said they were starting to ship it with all GEM applications. The salesman (and PC World) both stated that GEM/3 is much more faster than the previous version. And faster is better in my book. But, GEM didn't seem too awful slow so I can't imagine how it works now (that's ok ... surprise me). DR Customer Assitance indicated that Bitstream's FontWare automatically comes with their software if the software is bought from a DR dealer or is bought as a package deal, i.e., you buy a GEMAPP and the kernel together. They assured me that since I'm getting it from one of their dealers, I'd get this fontware stuff.

No, $315 isn't bad AT ALL!!! I'm not sure what my package will cost since they haven't called me back yet. While I said

Ventura is reasonable at $500. I don't want this all in one lump sum. I would like to keep the initial investment at or under $500. Now, if someone offered Ventura at an unbelievable low price, I might snatch it. I don't think that will happen though. Actually, I'll I'm wanting to do right now is to be able to draw up some block diagrams and put some text around it with some sort of font size control. I'm not looking for news paper font sizes and layouts and stuff like that, at least not yet. I might get Ventura later on if I need more power.

Fm Joe Kyle-DiPietropaolo: I just finished installing it, Shane. Good God.

The Bitstram software took not quite ten hours on my AT to generate a hundred and fifty font files for the screen and HP LJ+ printer. They take a little over a meg. If I had a HP LJ series II with full 300 DPI graphics capability rather than the Plus at 150 DPI, they would be four times that. Actually, it would use more space than that as I would build some fonts bigger than 30 point.

For the application you describe (block diagrams plus simple text), GEM Draw Plus by itself is quite adequate, we use it for that all the time. Print quality on a decent dot matrix printer is good too.

Fm Shane Dawalt: Joe, I was told the Tandy DMP 430 works just fine with GEM. (And the output look quite nice too.) I have run GEM on an ALPS P2000. Now that's class! I don't know if you've ever seen an ALPS printer before, but I have always wondered why they try to hide the paper from the user. The P2000 has 3 smoked plastic doors you must move to get down to the print head where the action is. In fact, to do any serious work, you have to slide 2 of the 3 doors off of the printer. It's not a big job, I just wonder why they find it necessary to have doors close over doors closing over doors. Actually, I think the printer looks QUITE nice. Oooo and is that dude heavy!

Back to GEM... **TEN** hours to generate 150 font files? You must be joking. Good God is right! Well, I guess that 24megs on drive C: isn't doing anything anyhow, why not fill it up with fonts and stuff?! I can't believe it took 10 hours. Mercy.

I don't think I'd want to attempt this just with GEM Draw +. See, I'm going to be drawing these block diagrams, but it will be inside of a document. GEM Draw + can only handle one page at a time. I need something which will allow word processing capabilities (I don't have a word processor ... well, none to speak of anyway) plus the graphics ability. GEM 1st WORD + allows 99 or 109 pages (forget which) plus you can insert line art or paint files. (Boy, paint files would do wonders for dot matrix wouldn't it?)

Fm Joe Kyle-DiPietropaolo: Shane, At least the font files only have to be generated once.

WordPerfect 5.0 will allow the integration of text and graphics. I should have mine within the next three weeks or so, and if the folks in Orem have done their usual good job, it may be all you need to use with GEM Draw+ for setting up the final document.

Fm Shane Dawalt: Joe, As I have heard before, WordPerfect OVER uses the function keys. I don't want to get a program which uses 3 or 4 layers of the same keys to do different things. Human Factors was not considered there. That PC-VT communications program I told you about (you know, the one which talks great with the VAX and UNIX systems) has that sort of command structure. What a pain.

Fm Joe Kyle-DiPietropaolo: Shane, Depends on the person, I guess. It has gotten to the point that my use of WordPerfect function keys is automatic. That is, I don't even look or think about what keystroke is required, my fingers just do it. Having a keyboard with the function keys on the left helps. In any case, 5.0 has a totally re-mappable keyboard. You want to put something somewhere else, go ahead and do so. Go ahead and put search-and-replace on ^Q^A if that makes more sense to you.

Bug in Mace's FORMATF

Fm MISOSYS, Inc: I came across a bug in one of the MACE utilities which brought to the attention of their Technical Support group. Here's what I had to say.

"Paul Mace Software, Attn: Technical Support, 400 Williamson Way, Ashland, OR 97520. Dear Tech Support,

I am using your FORMATF Version 1.3 formatting utility which is part of the Mace Utilities, version 4.10. I have recently tracked down a bug which is caused by the formatter. It seems that FORMATF does not put the correct value into the BOOT sector at relative position X'0B'. That location is supposed to contain the number of reserved sectors. For a floppy disk (standard), it should be a X'01' whereas FORMATF leaves it a X'00' value. On the surface, this appears to cause no harm until you either (1) attempt to use a utility which uses that value to find the first sector of the FAT, or until you try to use that diskette under some versions of MS-DOS 3.3.

Apparently, DOS 3.3 does indeed use that byte to determine the actual number of reserved sectors on a floppy; earlier releases of DOS do not and must assume a value of 1.

Because of this problem, we are forced to cease our use of FORMATF until you can provide us with a FIX."

We did receive a telephone acknowledgement of our report and Mace's staff advised me that it would be fixed in the next release. Just a word to the wise.

Microsoft WORD

A little on macros

Fm Roy Soltoff: One of the things I do when preparing program listings for TMQ is to replace the carriage returns (13D) with new lines (10D). That's because WORD treats a carriage return as a paragraph marker. Paragraphs have their own kinds of formatting. Although I can do just about anything to a group of paragraphs as I can to the lines within a single paragraph, there is at least one formatting facility that pertains only to a paragraph. That's the lined border.

A few folks have commented that the listings in TMQ were sometimes hard to read when presented in the snaking column format (that's a term to represent multi-column). So for this issue, I decided to enclose the listings in a box. The box is automatically generated by WORD when specified.

This paragraph is in a single line box.

On the other hand, I can also select a double lined box as in this paragraph.

The major chapter heading is enclosed in a bold-faced box. I won't show that here; I'm sure you get the idea. I can also select a line border by position.

Using line selections, I can specify that a paragraph is to be lined above and/or below as in this paragraph.

On the other hand, I also can select the lines to be to the left and/or right as in this paragraph. Actually, any combination of above, below, left, or right is permitted since each of the four directions is given a yes/no designation. You can also turn on the line drawing mode and use the ARROW keys to draw lines - all of this assumes that your printer is capable of reproducing line symbols. But that is getting away from what I really wanted to convey. Back to the subject of macros.

You should understand, that in order to surround a portion of text with a box, it is necessary to keep it within one paragraph. It can sometimes be accomplished through a group of connecting paragraphs by using lines. If you select left, above, and right, for the upper paragraph, and left, below, and right, for the lower paragraph, and left and right for all middle paragraphs, you can create the image of a box across paragraphs provided you keep the interparagraph separation at zero.

Rather than belabor the operation of enclosing multiple paragraphs with lines, which gets quite messy when considering a 100-line listing, I choose to convert the carriage returns to new lines and thereby tie the entire listing into a single paragraph. That job can be done manually. Within WORD, all you do is to position to the end of a line (with the END key), delete the paragraph marker with the DEL key, insert a newline with the SHIFT-ENTER key, then move down to the next line. One could also write a short program in C to do that prior to merging the listing into WORD. But for this issue of TMQ, I decided to write a macro in WORD to do that. WORD 4.0 added some very powerful macro capabilities which I haven't used to any great extent. Now was the time. Here's the macro which does this conversion.

```
«WHILE "1"="1"»
<end>
«IF selection<>"^p"»«QUIT»
«ELSE»
<shift enter><del>
«ENDIF»
«ENDWHILE»
```

Let me explain what this does. The two extreme statements specify a standard high-level language WHILE-ENDWHILE loop. That has the form of doing the range of tasks following the WHILE's condition until the ENDWHILE is reached based on the logical evaluation of the conditional test. In my macro, I use a standard "forever" loop by testing the conditional "1"="1", which is always true. The first task is the "<end>" statement which invokes the END key. This causes the WORD highlight (what non-WORD folks call the cursor) to be positioned at the end of the current line

Since my macro uses a forever loop, I have to have some other means of escape short of hitting the reset button (actually, you can escape from a macro's operarion by hitting the ESC key). This automatic means of escape is the **«IF selection<>"^p"»«QUIT»** test. That's part of a standard IF-ELSE-ENDIF construct found in languages uch as BASIC, C, and others. The word *selection* is a reserved word which designates the text within the highlight. I use that to test if the highlighted text is not a paragraph marker (the "^p" means paragraph marker to WORD). If it is not, the "«QUIT»" gets me out of the WHILE loop and the macro terminates.

On the other hand, if the test fails, that means the highlighted text is a paragraph marker. The **<shift enter><del>** inserts the newline character and deletes the paragraph marker (carriage return). The WHILE loop then continues by again testing its condition which is always TRUE and the IF-ELSE-ENDIF again executes. Macros are really like a high-level language. Although I used constants in the conditional tests, variables are permitted as well, and would be typically used.

Applications for the User

The following programs, although shown in their respective source code language, are nevertheless applications which may be directly usable by the non-programming user. All you need do is obtain the assembled/compiled program from the DISK NOTES 2.4 diskette which is associated with this issue of THE MISOSYS QUARTERLY.

SYSDRV/CMD
by David Goben

The following program is applicable to Model 4 TRSDOS 6.2 or LS-DOS 6.3 operation. It's purpose is to establish a drive other than drive :0 to be the SYSTEM drive, yet the original sequence of logical drive numbers 0-7 still matches the original physical assignments. The result is similar to Model III TRSDOS 1.3's MASTER command.

Here is an assembly listing to change system drives on 6.2 and 6.3, much like **SYSTEM (SYSTEM=d)**, except that logical assignments to drives are not changed (Drive :0 STAYS Drive :0). It is pretty thorough, and does plenty of tests to see if it can or cannot be installed. The only possible conflict is that it uses the 5-byte user storage area at X'0013'. I haven't seen it used anywhere, but if I can write a program that uses it, then so can someone else.

David Goben
67 Highland Road
Mansfield Ctr., CT 06250-1504

```
;SYSDRV 1.0 --- Copyright 1987 David Goben.
; Give away free at will. August 3, 1987
;This program sets a new system drive while
; maintaining the current logical drive assignments.
; Drive 0 will still be searched first for filespecs
; with no designated drivespec for saves and loads.
;
;Assemble to SYSDRV/CMD, and executing using:
;       SYSDRV :d
; where 'd' is the new system drive (BE SURE an
; Operating System is on the NEW DRIVE!  ).
;
;This program uses the 5 free bytes at
; X'0013' - X'0018' labeled USTOR$.
;I know of no program that uses this, but PLEASE be
; aware of this!!!!!!!!!!
;
        ORG     2600H
MAIN    LD      A,101           ;@FLAGS
        RST     28H             ;-- get flags to IY
        LD      A,(IY+27)       ;check DOS
        CP      62H             ;pre-6.2?
        JR      C,MAIN1         ;so skip CCKBRKC test
        LD      A,106           ;@CKBRKC
        RST     28H             ;abort if
        JR      NZ,ABORT        ;  BREAK pressed
MAIN1   PUSH    HL              ;input buffer pointer
        LD      HL,SIGNON$      ;say hello
        LD      A,10            ;@DSPLY
        RST     28H
        POP     HL              ;input buffer pointer
LOOP    LD      A,(HL)          ;get char stored there
        INC     HL              ;bump pointer
        CP      ' '             ;space?
        JR      Z,LOOP          ;ignore if so
        CP      ':'             ;drive separator?
        JR      Z,CHKDRV        ;yes
        CP      13              ;end of line?
;no, so check for drive number
        JR      NZ,CHKDRV+2
PERR    LD      HL,PERR$        ;report parm error
        DB      0DDH            ;hide next command
OLDDOS  LD      HL,OLDDOS$      ;pre-6.2 dos error
        DB      0DDH
NOTIN   LD HL,NOTIN$            ;SYSDRV driver not
        DB      0DDH            ; present error
NOTACT  LD      HL,NOTACT$      ;selective drive
        DB      0DDH            ; is not active
NODISK  LD      HL,NODISK$      ;no disk in target
        DB      0DDH            ; drive
NOTSYS  LD      HL,NOTSYS$      ;target drive not a
        DB      0DDH            ; system drive
IOERR   LD      HL,IOERR$       ;i/o error on new disk
        LD      A,12            ;@LOGOT
        RST     28H
ABORT   LD      HL,-1           ;error exit flag
        LD      A,(IY+27)       ;check for pre-6.2 DOS
        CP      62H
        RET     C               ;exit early if so
CKBRKC  LD      A,106           ;@CKBRKC
;clear break if present (no-job if pre-6.2)
        RST     28H
;return to DOS with good/bad flag in HL
        RET
;
;ensure that selected drive is acceptable
;
CHKDRV  LD      A,(HL)          ;get next char
        INC     HL              ;point to following
;(x-fer here if no ':' entered)
```

```
        LD      L,(HL)          ;save it
;drop ASCII offset from current char
        SUB     '0'
        CP      8               ;legal drive?
        JR      NC,PERR         ;no
        LD      (NEWDRV+2),A    ;save for later
        LD      H,A             ;else save it
        LD      A,13            ;check for ENTER
        CP      L
        JR      NZ,PERR         ;no, so error
        LD      A,(IY+27)       ;get OS version
        CP      62H             ;DOS < version 6.2?
        JR      C,OLDDOS        ;yes, dos is too old
        LD      A,H             ;Get new system drive
        OR      A               ;drive zero?
        JR      NZ,REL0         ;no
        LD      A,(13H)         ;else see if SYSDRV
        CP      0EH             ; driver present
        JR      NZ,NOTIN        ;not, so error
;
;do diagnostics on new disk
;
REL0    LD      C,H
;check selected drive for being active
        LD      A,40            ;@DCSTAT
        RST     28H
;is the new drive defined in the drive table?
        JR      NZ,NOTACT       ;no!
        LD      A,33            ;Do @CKDRV -- see
        RST     28H             ; if a disk on drive
        JR      NZ,NODISK       ;not present
        LD      A,81            ;@GTDCT -- point IY to
        RST     28H             ; new drive's table
        LD      D,(IY+9)        ;get dir cylinder
        LD      HL,1D00H        ;system buffer
        LD      E,L             ;set sector zero
        LD      A,85            ;@RDSSC --
        RST     28H             ; read system sector
        JR      NZ,IOERR        ;on target drive
        LD      L,0CDH          ;System flag in DIR
        BIT     7,(HL)          ;system disk?
        JR      NZ,NOTSYS       ;not a system disk!
        LD      L,0             ;reset buffer pointer
        LD      E,5             ;SYS1 on new drive?
        LD      A,85            ;@RDSSC
        RST     28H
        JR      NZ,IOERR
;get first byte of SYS1/SYS table entry
        LD      A,(HL)
;check for alive and system file
        AND     50H
        CP      50H
;Jump if not a system and/or not active
        JR      NZ,NOTSYS
;tests have passed, so set up new system drive
NEWDRV  LD      HL,000EH        ;set a LD C,<newdrv>
        LD      (13H),HL        ;set to lomem area
        LD      A,H             ;save new drive name
        LD      (0092H),A       ; to system FCB
        LD      HL,18BBH        ;init @DIRRD x-fer
        OR      A               ;If selecting drive 0,
        JR      Z,REL1          ; disable xtra routine
        LD      A,0C3H          ; else set JP address
        LD      (15H),A         ;set JP @DIRRD to
        LD      (16H),HL        ; extra routine
        LD      HL,13H
;
;insert new call into EXOVR routine
;over call to @DIRRD (or reset @DIRRD here)
;
REL1    LD      (1ABAH),HL
        LD      HL,SUCCES$      ;Report success if
        JR      NZ,REL2         ; not Drive :0
        XOR     A               ; else disable driver
        LD      (13H),A         ; presence indicator
        LD      HL,WTHDRW$      ;and report
REL2    LD      A,10            ;@DSPLY proper message
        RST     28H
        LD      HL,0            ;A-OK exit flag
        JP      CKBRKC          ;check break and exit
;
SIGNON$ DEFM    'SYSDRV -1.0- Set System Drive to
another Logical Drive'
        DB      10
        DEFM    'Written 1987 by David Goben.  For
free distribution'
        DB      10,13
PERR$   DEFM    'Parameter Error'
        DB      13
OLDDOS$ DEFM    'Need DOS 6.2 or higher'
        DB      13
SUCCES$ DEFM    'New SYSTEM drive installed'
        DB      13
WTHDRW$ DEFM    'SYSDRV driver now withdrawn from
system'
        DB      13
NOTIN$  DEFM    'SYSDRV driver not in system'
        DB      13
NOTACT$ DEFM    'Selected drive not enabled in DOS
system'
        DB      13
NODISK$ DEFM    'No disk present in selected drive'
        DB      13
IOERR$  DEFM    'I/O error in accessing new Drive.
Check and try again'
        DB      13
NOTSYS$ DEFM    'Selected drive' 's disk is NOT a
SYSTEM disk'
        DB      13
;
        END     MAIN
```

Searching with DSM4
by Charles A. Ainsworth

Introduction

DSM4 is a sort and search utility for model 4 random-access files, perhaps more versatile and powerful than may be realized by those who have only read the manual superficially. It works fast and is rich in valuable features.

According to the manual, there are two ways to run DSM4: One, in the immediate mode, by calling it from DOS and manually entering all sort and search parameters. The fact that it has so many valuable features necessarily makes this mode slow due to the numerous prompts and details involved and obviously cannot be used for repetitive or intensive work where speed is important. Also, although the instructions are

clear, it does take some pauses to handle one's input until one is familiar with the details.

The other way is to run it the first time around, for a given application, in the immediate mode upto the creation of a map file; subsequently, the procedure is repeatable by running a JCL which invokes DSM4 and the map file. Fast and easy!

However, I am developing an application in which the variable parameters to be entered are numerous to the point where it would be impractical to have a map file for every possible combination of input parameters, so I have experimented to reduce the number of map files required and to speed up processing setup as the application demands.

I use a number of random-access files containing strings, some of them as much as 28,000 records long, on 80-cylinder double-sided floppies, with LRL's from 5 through 23. My files are fielded for the length of each string plus 2 bytes for an index number, so I am handling strings from 3 to 21 bytes long. What I have to do is search through these files for strings of any of these lengths, with extensive use of the DSM4 wildcard character, "?". For example, I may search for a string like A??????S to find all eight-letter words beginning with A and ending with S.

As the combinations of numbers of letters, strings and wildcard characters are almost infinite and unpredictable, it becomes virtually impossible to have a fixed DSM4 map file for each specific search. In addition, as each word length is in a file of its own, a total of 19 files are involved; the file name is a parameter required by DSM4, so that would also multiply the number of map files.

Here are three alternative methods I have developed for my specific application; for all three, the search parameters are set up from an EnhComp-compiled CMD file, which speeds matters up somewhat, and the use of the XLR8er board promises to improve matters further still, in addition to giving me an extra drive.

Incidentally, the use of files compiled by EnhComp yields a bonus. If I were to use interpretive Basic, my system would have to load BASIC/CMD (24 K) plus my Basic program (10.5 K). EnhComp does it all with a single 19.5 K CMD file, saving on load time and on Ramdisk space which I need.

In all cases, there are two JCL's; the first, AUTOed from bootup, sets the system environment and calls the EnhComp CMD file to set up search and selection parameters. Once they are set up, the second JCL takes over, runs DSM4 and then calls another EnhComp compilation to display and/or print out the results of the search.

As I am not particularly interested in having the strings found by DSM4 in any sorted order, I have patched my copy of DSM4 per TMQ II.iii, p.57/58, and use the N command at the query for <A>scending or <N>o sort.

My examples are based on the simplest possible search, operating on only one ASCII field of each record. Those who have more complex search setups involving several fields should be able to develop a specific setup by extending my simplified examples. The main attributes for this are a good knowledge of the workings of DSM4 (read the manual!) and the patience to play around with map files to become familiar with them.

In addition, anyone wishing to experiment with these setups should bear in mind what is clearly stated in the DSM4 manual: If a JCL is used, there must be a map file.

METHOD NUMBER ONE

This procedure is initiated by a JCL, which sets up environment and system parameters and calls ZSHELL [a product previously sold by MISOSYS -ed] first and then the setup program.

The setup program then creates a sequential-access file with the parameters required by DSM4, to be used under ZSHELL, so parameters are fed into DSM4 at high speed. Once the parameter file is created, EnhComp exits to system and another JCL file is run; for example, if my parameter file is named "SEARCH" then the ZSHELL command from the JCL will be DSM4<SEARCH.

Note that this particular method does not use a map file at all, but operates in a way similar to the immediate one described in the Introduction, except that, instead of an operator entering parameters manually from the keyboard, ZSHELL takes over and uses a file to shoot the parameters at DSM4.

Incidentally, I also tried using TYPEIN/CMD, part of the LS Utility package, instead of ZSHELL, but couldn't get it to work, either because I was doing something wrong or perhaps because TYPEIN won't accept Break as part of the stream.

Here is an excerpt from my EnhComp-created CMD file for establishing search parameters:

```
760 PRINT TAB(19) "ENTER SEARCH STRING, MIN. 3
CHARS., MAX. 21"
770 PRINT TAB(25) "USE '?' AS WILDCARD
CHARACTER":PRINT
780 PRINT TAB(30) " ";:LINE INPUT A$:PRINT
790 A%=LEN(A$):IF A%<3 OR A%>21 THEN CLS:PRINT
TAB(23) "**** WRONG LENGTH ***, TRY
AGAIN":PRINT:GOTO 760
880 C$=STR$(A%)
890 IF A%<10 THEN GOSUB 1020 ELSE GOSUB 1030:
 REM REMOVE LEADING BLANK FROM C$
900 J$="DICT"+D$+"/RND":
 REM FILE TO BE SEARCHED.  D$=LEN OF SEARCH$
910 PRINT TAB(21) "INSERT BLANK SCRATCH PAD DISK IN
DR. 3":PRINT
920 PRINT TAB(18) "INSERT DISK WITH ";J$;" IN DRIVE
1 OR 2":PRINT:PRINT TAB(30) "<ENTER> TO
CONTINUE":LINE INPUT B$
```

```
930 N%=A%+2: REM N%=LRL=LEN (SEARCH$)+2
940 O$=STR$(N%)
950 IF N%<10 THEN GOSUB 1250 ELSE GOSUB 1260:
REM REMOVE LEADING BLANKS
960 W$=CHR$(13):REM  =<ENTER>
965 OPEN "O",1,"SEARCH:0"
970 PRINT#1,"I";J$;W$;P$;W$;"D";W$;"1";W$;D$;W$;
CHR$(128);"OS1";W$;"Y";W$;A$;
W$;W$;W$;"RSN";W$;"SEARCHER";W$;"N";W$;"3";W$
990 CLOSE 1
1010 END "DO = DSMSRCH/JCL":REM ONE ITEM OF JCL IS
DSM4<SEARCH OPERATING UNDER ZSHELL
1020 D$=RIGHT$(C$,1):RETURN
1030 D$=RIGHT$(C$,2):RETURN
1250 P$=RIGHT$(O$,1):RETURN
1260 P$=RIGHT$(O$,2):RETURN
```

Line 970 establishes parameters and places them in the SEARCH file; anyone wishing to understand it can simply call DSM4 from DOS and go through the motions, bearing in mind that the program establishes W$ as CHR$(13), i.e., ENTER, that J$ is the file name to be searched where the digit in the name represents the length of strings in the file, that CHR$(128) is BREAK as required at a certain point when entering parameters into DSM4, that P$ is the file LRL, that I want the temporary search file, if any, placed on drive 3, and so on.

I have not yet been able to test this particular method with an XLR8er board; with a model 4 running at 4 MHz, the parameters are fed into DSM4 very fast so one finds it hard to follow what's happening on screen. I don't know whether the higher speed from the XLR8er might perhaps be too high for DSM4 to handle.

METHOD NUMBER TWO

In this method, I created a map file for each particular file LRL and name. To create the 19 map files, I first did one, in the immediate mode, carefully noting the input to DSM4. I then copied this to an ALLWRITE file and easily and quickly created new lines to cover all the other map files with different LRL's and names. Then set up KSM with the resultant ALLWRITE file and created each map file by simply pressing Clear and the respective letter. I did something similar with the JCL files. BUT, naturally, I ended up with 19 map and 19 JCL files on disk. (I could have saved some disk space on the JCL's with PRO-PADS but there would have been little or no gain on the map files.)

One of the characters involved in creating such a KSM file is dec. 128, Break. ALLWRITE handles this easily with the special character feature, using Clear:128 but anyone using another text editor should check this. If a text editor cannot handle it, any other character could be substituted instead of Break and then changed to Break (hex. 80) with a disk file editor or by patching.

When I created the map files, I showed the search string as "???" for the 3 character strings, as "????" for the 4 character ones and so on. Then, when a search for a specific string is required, I edit the map file on disk which covers the pertinent string length and file name.

Here's what the program section, that changes the search string in the map file, looks like:

```
10 OPEN "R",1,"MAP8:0":
 REM MAP8= Map file for 8-character strings
20 FIELD 1, 255 AS A$, 1 AS B$
30 GET 1,2: REM The desired area begins in record
#2, position #83 @$for this type@% of map
40 MID$(A$,83,8)=D$:
 REM Changing a string of 8 "?" to search string D$
50 PUT 1,2:CLOSE
```

The 83 in line 40 is the beginning of the search string I found in my particular instance after a look at the map files with LSFEDII; a listing of the file in hex would have shown the same thing. Bear in mind that in the hex listing or with LSFEDII, counting is from zero whereas in a Basic or EnhComp program the count is from one. For other search and sort arrangements the starting point may be different.

METHOD NUMBER THREE

This method dispenses with the numerous map and JCL disk files of method number two by creating a map file from scratch for each particular search so there is only one map file and only one JCL to call it and DSM4. To set up this method, I created a set of map files in a manner similar to method two, one for each of my 19 files, not for inclusion in the system but only for study of the procedures for creating a one and only map file for each individual search which is discarded after use.

Here's what the map file for eight letter words (one of the 19) looks like for a search of file DICT8/RND searching for string A??????S (I have underlined significant data):

```
0000:00 =  4453344D41500000 0000000000000000
            D S 4 M A P . .  . . . . . . . .
0000:10 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:20 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:30 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:40 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:50 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:60 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:70 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:80 =  0000000000000000 0000000000000000
           . . . . . . . . .  . . . . . . . .
0000:90 =  0000000000000100 0000000000000000
```

```
                . . . . . . . . . . . . . . . .
 0000:A0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0000:B0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0000:C0 =  0000000000000800 0000000000000000
                . . . . . . . . . . . . . . . .
 0000:D0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0000:E0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0000:F0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:00 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:10 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:20 =  0000000000004449 4354382F524E4400
                . . . . . . D I  C T 8 / R N D .
 0001:30 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:40 =  0000000000000000 0A000000A63A0100
                . . . . . . . . . . . . . : . .
 0001:50 =  0100413F3F3F3F3F 3F53000000000000
                . . A ? ? ? ? ?  ? S . . . . . .
 0001:60 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:70 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:80 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:90 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:A0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:B0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:C0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:D0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:E0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
 0001:F0 =  0000000000000000 0000000000000000
                . . . . . . . . . . . . . . . .
(RECORDS 0002, 0003 AND 0004 ARE ALL ZEROS IN THIS
APPLICATION, PERHAPS NOT IN OTHERS)
 0005:00 =  0000000000000000 0000000000000100
                . . . . . . . . . . . . . . . .
 0005:10 =  0100010000010000 0000010008000D
                . . . . . . . . . . . . . . . .
```

This map file was created by a program, the pertinent excerpt of which is:

```
750 PRINT TAB(22) "SEARCH FOR WORDS IN DICTn/RND
FILES":PRINT
760 PRINT TAB(19) "ENTER SEARCH STRING, MIN. 3
CHARS., MAX. 21": REM MIN. AND MAX. STRING LENGTHS
USED
770 PRINT TAB(25) "USE '?' AS WILDCARD
CHARACTER":PRINT
780 PRINT TAB(30) " ";:LINE INPUT A$:PRINT:
 REM FOR THE EXAMPLE, A$="A??????S"
790 A%=LEN(A$):IF A%<3 OR A%>21 THEN CLS:PRINT
TAB(23) "**** WRONG LENGTH ***, TRY
AGAIN":PRINT:GOTO 760
880 C$=STR$(A%)
890 IF A%<10 THEN GOSUB 1020 ELSE GOSUB 1030:
REM ELIMINATE LEADING BLANK IN C$
900 J$="DICT"+D$+"/RND":REM NAME OF FILE TO BE
SCANNED.  D$= LEN (SEARCH$)
930 GOSUB 2500
940 GOSUB 3000
950 OPEN "O",1,"FIND/MAP:0"
960
PRINT#1,FO$;FG$;FB$;FF$;FK$;FH$;J$;FL$;FM$;FI$;FA$;F
P$;FB$;FC$;FB$;FC$;A$;FQ$;FD$;FD$;FE$;FS$;FB$;FC$;FB
$;FC$;FB$;FC$;FC$;FB$;FJ$;FB$;FC$;FK$;FC$
970 CLOSE
1010 END "DO = FIND/JCL"
1020 D$=RIGHT$(C$,1):RETURN
1030 D$=RIGHT$(C$,2):RETURN
2500 FB$=CHR$(1): FC$=CHR$(0): FD$=STRING$(240,0):
FE$=STRING$(224,0): FF$=STRING$(47,0)
2510 FG$=STRING$(144,0): FH$=STRING$(95,0):
FI$=STRING$(3,0): FJ$=STRING$(4,0): FK$=CHR$(A%):
FL$=STRING$((34-LEN(J$)),0)
2520 FM$=CHR$(A%+2):FQ$=STRING$((238-
A%),0):FO$="DS4MAP":FS$=STRING$(14,0)
2530 RETURN
3000 ON (A%-2) GOTO 3010, 3020, 3030, 3040, 3050,
3060, 3070, 3080, 3090, 3100, 3110, 3120, 3130,
3140, 3150, 3160, 3170, 3180, 3190
3010 FA$=CHR$(198):FP$=CHR$(42):RETURN
3020 FA$=CHR$(249):FP$=CHR$(10):RETURN
3030 FA$=CHR$(144):FP$=CHR$(21):RETURN
3040 FA$=CHR$(172):FP$=CHR$(1):RETURN
3050 FA$=CHR$(124):FP$=CHR$(1):RETURN
3060 FA$=CHR$(166):FP$=CHR$(58):RETURN
3070 FA$=CHR$(55):FP$=CHR$(1):RETURN
3080 FA$=CHR$(29):FP$=CHR$(1):RETURN
3090 FA$=CHR$(7):FP$=CHR$(1):RETURN
3100 FA$=CHR$(244):FP$=CHR$(0):RETURN
3110 FA$=CHR$(228):FP$=CHR$(0):RETURN
3120 FA$=CHR$(214):FP$=CHR$(0):RETURN
3130 FA$=CHR$(38):FP$=CHR$(11):RETURN
3140 FA$=CHR$(190):FP$=CHR$(0):RETURN
3150 FA$=CHR$(180):FP$=CHR$(0):RETURN
3160 FA$=CHR$(171):FP$=CHR$(0):RETURN
3170 FA$=CHR$(163):FP$=CHR$(0):RETURN
3180 FA$=CHR$(155):FP$=CHR$(0):RETURN
3190 FA$=CHR$(149):FP$=CHR$(0):RETURN
```

Again, I won't explain all the details of this program; it was developed following the location of the various parameters in the experimental map files; for example, on the sample map file listed above in hex., 00:00 "DS4MAP" is constant; there are then 144 double zeros; followed by 01 (sort field, I believe); followed by 47 zeros; followed by the length of the sort string. The LRL is the 0A in line 0001:40. And so on.

These figures are developed in subroutine 2500, directly relatable to the sort parameters, and anyone wishing to play with this is advised to create printouts of test map files as guidance in writing the program. There was one part I stumbled on, not being an expert assembly-language programmer <blush!>. I could not decipher what some of the figures in the map files apply to, so used my "quick and dirty" method and added subroutine 3000 simply copying the specific values for each sample map file.

Remember that CHR$(n) uses decimal values but that one is dealing with hex. values when listing a map file to screen or to a file editor.

Tests on this method 3, using a setup file compiled with EnhComp and the XLR8er board, (not up to full working speed as I haven't yet installed the recommended 150ns motherboard RAM chips) show that from the moment the test string is entered, such as A??????S, total time for searching through some 15,000 records and commencing the screen display of what has been found, is about 45 seconds, including loading my CMD file (19.5K), DSM4 (34 K) and DSORT (4.5 K) from Ramdisk, plus the time to locate my random-access file in one of 4 floppies, which I feel is pretty good. Creation time for the map file (lines 790 through 1010 of my program) is about 1.5/2 sec. with the XLR8er active.

A Tidbit re DSM4

As mentioned in the introduction, my system goes automatically through the mechanism of setting up search parameters and the search itself, ending with display and/or printout of matching strings which were found. The latter operation is based in general lines on the system described on page 14 of the DSM4 manual, which uses the file created by DSM4/DSORT as an index to access the user's data file, as follows (I quote from the manual):

```
10 OPEN "R",1,"MYINDX/IND",2  'Open the index file
(Lrl of 2)
20 FIELD 1,2 AS S$ 'Field the index file
30 OPEN "R",2,"MYDATA/DAT",lrl 'Open data file with
proper LRL
40 FIELD 2, xxxxxxxx 'Establish fields in the data
file
50 FOR L%=1 TO LOF(1)  'For as many records as are
indexed
60 GET 1,L% 'Get the record number from the index
file
70 GET 2,CVI(S$) 'Get that record from the data file
80 'Do whatever processing is required (e.g. print
the records)
90 NEXT L% 'Continue with the next one
100 CLOSE
110 END
```

That works fine, BUT it may happen, on a particular run, that DSM4/DSORT find NO ITEMS which match the search and sort parameters, in which case record #1 of MYINDX/IND (line 10) would produce, in line 70, CVI(S$)=-26215. This, naturally, causes an error which throws one out of the program, and perhaps some confusion and lost time.

In my own use, I insert two extra lines, which, worked into the above program, would be:

```
25 GET 1,1:IF CVI(S$)=-26215 THEN CLOSE:GOTO 120
120 PRINT "NO MATCHING ITEMS FOUND":GOTO nnnn: REM
nnnn=LINE TO START ANOTHER SEARCH OR CONCLUDE
PROCESSING
```

Update from MISOSYS

After receiving Charles' article, something in my mind recollected coming across an old LSI note on the DSM "/MAP" file. I dug through some old material and came up with the following. Obviously, this information would have saved Charles a little bit of time, but at least now it will be more widely available. Here's the scoop:

DESCRIPTION OF A DSM "/MAP" FILE

Map files for DSM4 and DSM5l are almost identical even though the allowable number of output fields are different. Following is a description of the map file, which should be adaquate to create such a file from an independent program.

ALL 2 byte fields are store on disk as LSB, MSB. From BASIC, this type of storage is the same as MKI$ used with random files. String fields (filename and select string) must have unused positions filled with binary zeros. ALL unused data items should be written as binary zeros. A description such as 2*24 means that there are 24 consecutive data items, each being a 2 byte integer. Input field starting positions are offset from 1 (i.e, if the starting position is byte 18 write the hex bytes 12 00 to the file). The delete char position is offset from 0 (i.e, if the position is 1, write the hex bytes 00 00 to the file).

Relations are 0=EQ, 2=GE, 4=LE, 6=GT, 8=LT, 10=NE. Connective values are 0=None (last output field), 1=AND, 2=OR. For a 2 byte field described as always 0, write a 00 00 to the file; if described as always 1, write a 01 00 to the file.

The section containing the output information has 24 fields each containinging 37 characters. Even though DSM5l does not allow for 24 output fields, all 24 positions must be written, with the unused positions filled with zeros.

Bytes Description

6	Header ("DS4MAP" for DSM4, "DS3MAP" for DSM5l) 144 Always zeros
2*24	Starting position of input field (offset from 1) 2*24 Length of input field
2*24	Type of input field
32	Filename (and extensopn, drive if used) of data file. 2 Zero
2	Lrl of data record
2	0
2	Total records in data file

This next section contains the select descriptions

37*24	2 bytes - always 1 2 bytes - input field number (offset from 1) 33 bytes - select string
1*24	Output field relation
1*24	Output field connective
1*24	Sort flag (1 if attach to sort, 0 if not)
2	Always 1
2	Always 1
2	Total input fields
2	Delete character (Zero if none) 2 Delete relation
2	Delete start position (offset from 0)
2	Output field count 2 Length of sort string

Total = 1310 bytes

Testing printer ready from BASIC by Dick Hollenbeck

Perhaps the old German saw that says the farmer refuses to eat anything he doesn't know, partially explains my penchant for blaming whatever software package is currently on-line whenever I encounter a glitch. Although my knowledge of coding is sketchy, my understanding of the finer points of computer hardware is non-existent.

Case in point: Ever since getting my Model 4 in 1984, I have used a short BASIC routine that polls port 248 to determine whether the line printer is on line. This undocumented routine sends output to the line printer if it is on line, otherwise, output goes to the screen. The code looks like:

```
10
.
80 PRINT"For Hard-Copy turn on line printer"
.
350 LP=((INP(248)AND 248)=48)
.
430 IF LP THEN LPRINT"Some good stuff and whatever"
.
```

I think you get the idea. I'm not sure who first found this little routine, but I've found it very handy. Recently, I was updating a couple of home rolled applications under LS-DOS 6.3 when all of a sudden it refused to work properly. I immediately concluded the glitch was some change in the DOS software. So I got off a letter to LSI explaining my problem and asking if they knew of a fix. A few days later, Virgil gently suggested that I look at the hardware environment - especially interconnecting cables. There it was! (God bless you guys that work with the public.) I use three printers and had accidentally broken one of the connector teeth. A new cable fixed the problem.

In his reply, Virgil did not seem to be too familiar with this little routine, and I thought that if he wasn't up on it, maybe some of you other pros also may not be aware of it. Over the past four years I've received a lot of help, and sending this along may be partial repayment for all that help.

The DEMO/BAS routine is intended to be run from BASIC. Suggest that it be run twice: First with your line printer off line, then with it on line. The DEMO/CMD routine was generated by EnhComp, using DEMO/BAS as the source file. Suggest it be run twice, also.

Dick Hollenbeck
10440 Crete Dr
El Paso, TX 79924-1720

```
10 ' *** Demo Program to determine whether to
produce Hard Copy ***
11 '
12 'By polling port 248, BASIC is capable of
determining whether the
13 'system Line Printer is on-line.  This is useful
when the operator
14 'wishes to produce hard-copy output.  Depending
upon how the segment
15 'is coded, the programer can let the operator
elect to send output
16 'only to the screen, only to the line printer, or
to both.  The basic
17 'statements are as shown -
18 '
19 '   100 PRINT"For Hard Copy - turn on Line
Printer"
20 '    .
21 '    .
22 '    .
23 '   300 LP=(INP(248)AND 240)=48)
24 '    .
25 '    .
26 '    .
27 '   500 IF LP THEN LPRINT"Whatever - so on and so
forth"
28 '
29 'Line 100 prompts the operator; 300 determines
whether the operator took
30 'action for hard copy, and if he did, line 500
produces the desired
31 'hard copy output.
32 '
33 'This scheme works under all TRSDOS 6 versions,
LS-DOS 6.3, and with
34 'EnhComp.
35 '
36 'The demonstration code follows -
37 '
38 '
```

```
100 BANNER$=" > > > > > > > > > >   LINE PRINTER
DEMONSTRATION   < < < < < < < < < <"
110 SCRNFIL1$="This is Line # ":SCRNFIL2$=" of
LPRINT Demo"
120 PROMPT$="For HARD COPY turn on Line Printer -
Press <ENTER> when ready  "
130 CLS:PRINT BANNER$+STRING$(5,CHR$(13))
140 Q$="":PRINT PROMPT$;
150 Q$=INKEY$:IF Q$=""THEN 150ELSE IF
Q$<>CHR$(13)THEN 150ELSE CLS
160 LP=((INP(248)AND 240)=48)
170 PRINT BANNER$:PRINT:FOR X=1 TO 20
180 PRINT SCRNFIL1$;X;SCRNFIL2$
190 NEXT X
200 IF LP GOTO 240ELSE FOR X=1 TO 1500:NEXT X
210 CLS
212 PRINT BANNER$+STRING$(10,CHR$(13))
214 PRINT TAB(25)"Demonstration
Concluded":PRINT:PRINT
220 PRINT TAB(12)"Press <BREAK> to re-gain control
of your system ";
230 GOTO 230:END
240 LPRINT TAB(7)BANNER$:LPRINT:FOR X=1 TO 20
250 LPRINT TAB(20)SCRNFIL1$;X;SCRNFIL2$
260 NEXT X:GOTO 210
```

Tapes, Disks, and CMD Files
by Gary W. Shanafelt

or, How To Convert Your Favorite Special-Format Programs to Normal CMD Files

Gary W. Shanafelt
1197 Grand Ave.
Abilene, TX 79605

Games for the TRS-80 were once a hot commodity. Though the game craze peaked in 1983, every so often people still seem to want to transfer a tape program to disk, or to fix a program on a self-booting disk so that it can be backed up. Transferring programs in BASIC from tape to disk was generally a piece of cake; but games in Z80 machine language often represented a real challenge. At one point, making such conversions of machine code was also considered rather reprehensible, for a good deal of it was done by pirates who then gave away the programs to all and sundry. It is still an open question how much such practices hurt software makers then (and continue to hurt software makers today). As far as the TRS-80 world is concerned, though, it's pretty much a dead issue. Most programs still on the market are not protected, for good or ill; and most of the makers of the games that were protected are no longer in business. In fact, if you have a protected program that somehow gets destroyed, with the manufacturer defunct you will never be able to replace it -- unless you have it in a form that is easily copied. And if you can make backups of self-loading games with utilities like Copycat or Super Utility, you still can't put them on any medium except their original tapes or dual-density disks -- not a hard disk, not on the double-sided disks of the Model 4D, not on the new 3" drives which a few people have tried to hook up to their TRS-80's.

Why can't the great machine-language games of the TRS-80 world be put into normal CMD format so that they can go on any kind of disk and be easily preserved? Generally, they can, though it takes a certain amount of effort. With a Model 4, practically any game can be converted to a regular DOS format unless it is an adventure game like Forbidden Planet with constant disk accesses. (This type can probably be converted as well, but I have never tried to do it).

The easiest programs to convert are tapes with no special loaders. If it can run on a Model 4 in III mode, it can be put on disk (some early Model I tapes will not run on the Model III). Just use CMDFILE on your LDOS 5.3 disk and follow the instructions. Tape games generally overlay the area of memory occupied by the DOS, so they crash if loaded directly into the computer from a disk. CMDFILE not only transfers such files from tape to disk; it also changes or offsets the program's memory load addresses to those above the DOS, then adds a machine language routine at its end so that once the program is safely loaded into the new addresses, it is then automatically shifted back down over the DOS (which at that point is no longer needed for anything). Rally or Caterpillar are good examples of this type of game. With CMDFILE, they transfer from tape to disk in a matter of minutes.

A fair amount of machine-language tape games will transfer to disk with CMDFILE, yet crash at some point when they are run. Big Five's Attack Force, for example, transfers easily from tape to disk, yet freezes when you get to the high scores (ironically, while it does this under LDOS, it runs fine with TRSDOS 1.3, which may be the only good thing to be said for that benighted operating system). Similar things happen with Cornsoft games like Bounceoids or Frogger. The problem is that these games make references to the memory just above 4000 (hex). They expect it to be initialized for tape operation, and if the DOS has altered it, they hang up. Luckily, the Model 4 provides a rather easy way out of this difficulty.

Because the Model 4 DOS does not load in the same memory area as the Model III DOS's, it is possible to load a game in Model III mode and then reset with LS-DOS 6.3 without overlaying the game memory. It remains intact to be examined with DEBUG or to be dumped to disk. To get a tape game like Attack Force to work properly, do the following: (1) Try to transfer it from tape to disk with CMDFILE. Make sure you write down the ending address and the transfer address. You will need both of them in a minute. (2) Once you've determined that the game, now on disk, still doesn't work properly, turn off your computer to clear all the memory. (3) Turn it back on, go to Cassette BASIC, and load and run the program. (4) Slide in your trusty LS-DOS disk and reset. (5) Dump the memory from 4000 (hex -- all addresses here will

be in hexidecimal) to the ending address you wrote down in step one. Use the transfer address you wrote down then. (6) Return to Model III mode. Use CMDFILE to offset the dump you just completed. The result will be the same thing you would get in a straight transfer from tape, except that the game begins at 4000 (or will when the offset moves it there from high memory). The tape-initialized routines the program expects to find have been tacked on to its start. The program will be maybe 1K longer than it would be otherwise, but with the extra memory from 4000 to its original starting address, it will now run just as if it had been loaded from tape.

The hardest category to transfer are games with self-loaders. CMDFILE won't work on these. To get the game onto disk, you have to make a straight dump from memory. (1) Start out as above, turning off your computer to clear the memory, then going to Cassette BASIC and running the program from tape. (2) Reset with LS-DOS 6.3. (3) You now have to determine the start, end, and transfer addresses of the game. I've simply dumped the whole memory from 4000-FFFF to MemDisk (I have a 128K machine) and then examined it with LS-FEDII. Start at 4000 and work your way up. Most cassette games end around 8000, since they assumed no more than 16K memory. You can tell you've reached the end when you start seeing sector after sector with nothing but 00 FF 00 FF 00 FF patterns. (4) Unless there is a large amount of the same pattern at the beginning, assume a starting address of 4000. The ending address is wherever the 00 FF pattern resumes. (5) Finding the transfer address is more difficult. Most games start with the code F3 to disable the interrupts, and a 31 to set the stack pointer is another sign that you're close to the entry point. Use LS-FEDII's search feature to locate the code sequence F3 31; if there's only one occurrence, try this as the transfer address. (4) Return to LS-DOS and load your memory dump (use the "load" command; do NOT try to run it). (5) Dump the memory from 4000 to the ending point you determined; use your likely transfer address. (6) Go back to Model III mode and CMDFILE your final dump, adding an offset to it.

Now, if you were lucky with the transfer address, the program will run. If not, you'll have to try other addresses, perhaps every F3 instruction, until you get the right one. This can be pretty tedious. Sometimes looking for a command to clear the screen (loading spaces starting at 3C00 or a call to 01C9) will help. Luckily, games by the same author often use the same subroutines, so if you figure out one you have a key for the others. All five FunSoft games have the same command sequence at their entry points.

If you're trying to convert a self-loading disk to CMD format, you follow pretty much the same procedure as for protected tapes. There are two main differences: (1) Disk versions generally load above 5200 so there is no problem with overlaying the DOS (code around 4000 indicates their loaders and can usually be disregarded). With a starting address above 5200, once you have a dump you don't have to worry about an offset. (2) Disk versions usually save the scores back to disk, but to a specified track and sector -- not in DOS file format. These routines should either be disabled or, if you're really ambitious, be replaced with routines that create a normal DOS data file. Often, a call instruction to an address in the 4000's at the start of the program and at the high scores roster jumps to a save-score routine in the loader. Zero these calls or add your own subroutine here.

Even supposedly "32K" games aren't much longer than 16K. Sometimes the only difference between tape and disk versions of a game is a score-saving routine, though often digitalized voices were added (one wonders what could have been done with an arcade game that utilized the computer's full memory: imagine a 48K version of Sea Dragon!) Adding a save-score routine to a tape program transferred to disk often makes it functionally identical to the disk version. This conversion can be done even if the program overlays the DOS. The only reason to go to this much trouble is if you have a tape version but now find it impossible to locate a disk version. Using the game Laser Defense as an example, I'll explain what is involved.

The tape version of Laser Defense is protected and it loads in low memory. I does not have a self-loader, though, so you can use CMDFILE to determine the transfer address. Beyond that, you use the procedure outlined above for a protected file, running the program from the tape and then dumping everything from 4000 to where the data in the memory ends (which is higher than CMDFILE says it will be). The relevant addresses, at least on the version of the game that I bought and worked with, are 4000-7D04, transfer at 5200 (where you will find the sequence F3 31 as mentioned above). Once you have the dump, you return to CMDFILE to add an offset. I used an offset of 7000, which makes it easy to keep track of memory addresses: each address in the offset version is 3000 higher than the actual address it will end up at when the game is loaded. Once this is done, you have a working tape version on disk. Load the game from the disk and play it through a few times to make sure it works properly. If you made a mistake somewhere, it will hang up or reboot either at the title page or at the end of a game sequence when you try to record the high scores.

The save-score routines are the same as those you would add to any disk game. There are three additional considerations, though, because the program will overlay the DOS when it runs.

(1) You have to add another routine which moves the memory from 4000-5200 (the area of the DOS) to high memory BEFORE the offset routine goes into gear. When the program saves the scores, you move the program area between 4000-5200 to another area of high memory, then move the DOS back to 4000 for the score-saving. Once the scores are recorded, the lower part of the program is moved back down to its normal place. The routines to do this are quite short, but you obviously need a lot of free memory for moving

everything. You'd better have at least 48K ram in Model III mode...

(2) Keeping track of the address where the scores are stored in memory can get rather tricky. At the start of the game, previous scores are loaded from disk to the address of the program in its offset position in memory (that is, before the offset routine moves it down to 4000). However, when a new score is saved, it is sent back to disk from the original (unoffset) address. It that address is below 5200, then the score-saving routine has to find the score at wherever you've moved the lower part of the program in high memory when you switched it with the DOS. If this seems too confusing, you can set up another part of free memory as a scores buffer and do all your loading and saving from there. With Laser Defense, the scores are stored at 7700, which is A700 if the program is offset 3000 bytes (starting at 7000 instead of 4000), so a separate scores buffer is really not necessary. But I will include one just to show what is involved.

(3) The original stack of Laser Defense is located in the memory below 5200. When you start switching chunks of memory, it gets disrupted and the program will crash when it tries to save the scores. So, the stack pointer has to be set to some area of high memory where it will cause no problems. The stack pointer is located at 5201 (the 31 command).

All this may sound rather confusing, so if it makes no sense refer to the end of this article where a brief table of the routines and the sequence in which they take place is given. The hardest part is not the writing of the new routines, but locating where in the program the high scores are recorded in the memory before the program jumps back to its beginning. That can take as long as finding a transfer address. And of course there's no extra memory waiting there to hold a score-saving routine, so you have to jump to the end of the program for it. Since the jump command takes up 3 bytes, you also have to put the bytes that used to be where your jump command now is at the start of your routine. In Laser Defense, the scores are in a subroutine that ends with a call followed by a return code. You put your jump where the call used to be, zero the return code, and then add the call at the address you are jumping to, with the score-saving routine immediately after it and a return command after the score routine. Luckily, the routine to load the scores is much easier to locate: you simply put it at the start of the program.

If you've gone this far, there's one more change you can give to Laser Defense (or any other game) which is especially nice if you want to put it on a hard disk. As practically all games stand now, there is no way to return to DOS short of rebooting the computer. Since some hard drives require you to have the DOS on a floppy every time you boot, switching from one game to another can involve a fair amount of disk switching and waiting. A simple command can rectify that. From the title page of Laser Command, you are told to press "S" to start the game. The code uses the two instructions 3A0438 CB5F to scan the keyboard memory and check when the letter "S" is pressed. Simply find that code in the program and insert a jump to the end of the program over the first three bytes. Now, at the end, you add a command to scan also for the letter "X" to "exit" (or whatever other key you want to use). When "X" is pressed, the program moves the DOS from its storage place in high memory back down to 4000 and then jumps to 402D, the normal exit point to DOS. This is also easier listed than described; see the routine at the end.

How, precisely, do you actually do all this? I've just added an extra load module of 256 bytes to the end of the game, put the transfer address beyond it, and then typed in the routines by hand (who needs mnemonics?). This is rather easy with LS-FEDII, whose "I" and "J" commands allow you to check constantly for the accuracy of your instructions and memory addresses. I suppose this could also be done with DEBUG, but I've never tried it. It certainly wouldn't be as easy. LS-FEDII is without doubt the nicest debugging and zapping tool I've ever come across, and projects like this would be immensely more difficult without it.

General Memory Map of converted LASER DEFENSE

Bulk of the program, as dumped from memory

The offset module or routine, added by CMDFILE

The score and exit routines added as described above

At the end of the routines the name of the score data file, here called LASER/HIS. Make sure you put a carriage return (0D hex) after the name

Finally, the transfer module (02 02 + transfer address, now set to transfer to the new load-score-from-disk routine)

THE ROUTINES IN THE ORDER THAT THEY ARE USED

The program loads at 7000, the offset address. Then control is transferred to the load scores routine:

```
F3              Disables interrupts
2100E8          Clears area at E700 to be used as
                scores buffer
11FFE7
010001
3620
EDB8
2100E7          Start of scores buffer
11BAAD          Location of LASER/HIS
0600
CD2444          Opens scores data file
2010            Jumps to close file routine if error
                (ie, no scores file on disk if this
                is the first time the game is being
                played)
CD3644          Reads the file from disk to memory
200B            Jumps on error
2100E7          Moves the scores from the buffer to
                their location in the offset  version
                of the program
```

```
1100A7
012800          28 (hex) bytes, the length of the
                scores
EDB0
11BAAD          LASER/HIS
CD2844          Closes the file
F3
210040          Moves the DOS memory up from 4000 to
                D000
1100D0
010012
EDB0
C305AD          Jumps to the start of the offset
                routine added by CMDFILE
```

Here is the offset routine:

```
F3
210070          Moves the original program from 7000
                to 4000 (note that the new routines
                are NOT moved)
110040
01053D
EDB0
C30052          Jumps to original transfer address
                (note that the stack pointer at 5201
                has to be changed; I zapped the
                instruction to 3100C8).
```

The game now runs its course until you get destroyed. If you make a high score, you type in your initials and press <ENTER>. Then the program jumps to the added save-score routine. The routine begins with the call which used to be where the jump instruction now resides. Then follows the new routine itself:

```
CDF75F          Original call instruction, displaced
                here by instruction to jump to this
                routine
F3
210040          Moves the lower part of the program
                from 4000 up to B000
1100B0
010012
EDB0
2100D0          Moves the DOS from D000 back down to
                4000
110040
010012
EDB0
210077          Moves the scores from their location
                in the unoffset program to the scores
                buffer
1100E7
012800
EDB0
2100E7          Start of scores buffer
11BAAD          LASER/HIS
0600
CD2044          Initializes a file
C0
CD3944          Writes the scores to the file
11BAAD          LASER/HIS
CD2844          Closes the file
F3
2100B0          Moves the lower part of the program
                from B000 back down to 4000
110040
010012
EDB0
C9              Returns to the program
```

Finally, the "Exit to DOS" routine. By adding a jump instruction where the scan for "S" is located, you displace that instruction to the end of the program. The routine looks like this:

```
3A0838          Keyboard scan for "X"
CB47
2006            Relative jump to exit routine below
                if "X" is pressed
3A0438          Keyboard scan for "S" (moved here to
                make room for jump to this routine)
C36255          Jumps back to next instruction in
                body of program
F3              Disables interrupts and begins exit
                routine
2100D0          Moves DOS from D000 back down to 4000
110040
010012
EDB0
CDC901          Clears the screen
C32D40          Jumps to DOS
```

The Programmers' Corner

WAMDUMP
by Claude E. Hunter

I received PRO-WAM version 2 several days ago and find the package very interesting. I'm including a small program (nothing elegant) I wrote that will take a PRO-WAM /APP file and dump it as a regular command file. This will allow the /APP module to be easly disassembled by PRO-DUCE. To invoke it, just issue the command:

"WAMDUMP module:#"

Do not include an extension - WAMDUMP will append /APP automatically.

Claude E. Hunter
202 Rendova Road
Fort Ord, CA 93941

```
;WAMDUMP Version 1.0.0 PROWAM Application Command
Dump Utility
;Copyright 1987 by Claude E. Hunter
;
;Placed in the Public Domain.
;
;Extract PRO-WAM application using WAMLIB
;
;Issue WAMDUMP filename:d
;'
;NOTE: Do not use an extension.  WAMDUMP will
automatically open the
;       application with the /APP extension and write
the Command file
;       with an /CMD extension.
;
;PRO-DUCE can now be used to disassemble the PRO-WAM
application.
;
        ORG 3000H
START   PUSH HL                 ;Save command line
        LD      HL,TITLE        ;Display title
        LD      A,10
        RST     40
        JP      NZ,ERROR
        POP     HL
        LD      DE,FCB          ; Point to FCB
        PUSH    DE
        PUSH    DE
        POP     IY              ; and move to IY
        PUSH    HL              ;Save filename
        LD      A,78
        RST     40              ;Move filename to FCB
        JR      NZ,ERROR
        LD      HL,APP          ;Append /APP ext
        LD      A,79            ; to filename
        RST     40
        LD      B,0
        LD      HL,BUFFER
        LD      A,59
        RST     40              ;Open file
        JR      NZ,ERROR
LOOP    LD      A,67
        RST     40              ;Read record
        JR      NZ,EOF          ;Jp if error
        LD      A,(IY+4)        ;Bump buffer to load
        INC     A               ; all of record
        LD      (IY+4),A
        LD      A,(COUNT)       ;Bump num of records
        INC     A               ; loaded
        LD      (COUNT),A
        JR      LOOP            ;Loop for next record
EOF     CP      28              ;EOF error
        JR      Z,EOF1
        CP      29              ;Rec num out of range
        JR      Z,EOF1
        JR      ERROR
EOF1    LD      A,(COUNT)       ;Get record count and
        DEC A                   ; decrement for last
        LD      (COUNT),A       ; record
        LD      A,(IY+8)        ;Get EOF offset
        LD      (EOFOFF),A
        LD      A,60            ;Close /APP file
        RST     40
        JR      NZ,ERROR
        POP     HL              ;Restore filename
        LD      A,78
        RST     40              ;Move filename to FCB
        JR      NZ,ERROR
        LD      HL,CMD          ;Point to /CMD ext
        LD      A,79            ;Add /CMD ext
        RST     40              ; to filename
        LD      HL,BUFF
        LD      A,58
        LD      B,0
        RST     40              ;Open /CMD file
        JR      NZ,ERROR
        LD      DE,FCB
        LD      HL,BUFFER
        LD      C,5             ;Load module header
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      C,0             ;Load module header is
        LD      A,4             ; 256 bytes
        RST     40
```

```
        JR      NZ,ERROR
        CALL    WRITE           ;Write load module
        LD      A,(COUNT)       ; header
        DEC     A
        LD      (COUNT),A
        CALL    EXECUTE
        JR      Z,EXIT
ERROR   OR      128             ;Set high bit
        LD      C,A
        LD      A,26
        RST     40
EXIT    LD      HL,0
        LD      A,22
        RST     40              ;Exit
LOAD    LD      C,1             ;Load block
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      C,2             ;Load 258 bytes
LOAD1   LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      C,0             ;Load address
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      A,(ADD)         ;Load address
        LD      C,A
        INC     A
        LD      (ADD),A
        LD      A,4
        RST     40
        JR      NZ,ERROR
        RET
WRITE   LD      B,0             ;Write 256 bytes
WLP     LD      A,(HL)          ;Get byte from buffer
        LD      C,A             ; and send to file
        LD      A,4
        RST     40
        JR      NZ,ERROR
        INC     HL              ;Bump buffer
        DJNZ    WLP             ;Loop until 256
        RET
LAST    LD      C,1
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      A,(EOFOFF)
        INC     A
        INC     A
        LD      C,A
        CALL    LOAD1
        LD      A,(EOFOFF)
        LD      B,A
        JR      WLP
TRA     LD      C,2
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      C,2
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      C,0
        LD      A,4
        RST     40
        JR      NZ,ERROR
        LD      C,40
        LD      A,4
        RST     40
        JR      NZ,ERROR
        RET
```

```
EXECUTE EQU $
;       Write load info to cmd file
LP1     CALL    LOAD
;       Write 256 block to file
        CALL    WRITE
        LD      A,(COUNT)       ;Get blocks to write
        DEC     A               ; and dec
        LD      (COUNT),A       Loop
        JR      NZ,LP1          ; if not last block
;       Write last part of last block
        CALL    LAST
        CALL    TRA             ;Write transfer add
        LD      DE,FCB
        LD      A,60
        RST     40              ;Close file
        RET
TITLE   DB      'WAMDUWP Version 1.0.0 PRO-WAM
Application Command Dump'
        DB      ' Utility',10
        DB      'Copyright 1987 by Claude E.
Hunter',10,10
        DB      'Placed in the Public Domain.',13
APP     DB      'APP',13
CMD     DB      'CMD',13
ADD     DB      40              ;Start load address
EOFOFF  DB      0
COUNT   DB      0
FCB     DS      32
BUFF    EQU     3200H
BUFFER  EQU     3300H
        END     START
```

EXMEM <-> BASIC INTERFACING

By Mark A. Mueller
CIS 73100,2413

Here is an article for TMQ explaining a technique I use to access extended memory from BASIC on a Model 4 through EXMEM. It is written on a fairly novice level, and I hope it will be suitable.

Mark A. Mueller
613 S. LaGrange Rd.
LaGrange, IL 60525

One of the more annoying problems with BASIC on the model 4 under LS-DOS 6.3 is the tight memory situation when even a couple of filters or drivers are in high memory. Compound that with a program that uses lots of array space, and the familiar "out of memory" message greets you. EXMEM has come to my rescue, giving my programs a minimum of 57K effective memory, which is more than enough (so far).

So, how does a programmer get at all this extra memory? The BASIC in LS-DOS 6.3 implements an incredibly powerful function which accesses the operating system supervisor calls (SVC's). This access is fast, efficient and very easy to use, and complements EXMEM very nicely. EXMEM, in case you missed the last few TMQ's, allows access to alternate banks of memory (those above the "normal" 64K) in 256 byte pages. It works with the other 64K in a "stock" 128K Model 4 and with other optional memory add-ons like the H.I.Tech XLR8er or the Alpha Tech memory boards. With EXMEM installed, the @EXMEM SVC can be used through the USR11 BASIC interface to store and retrieve data in these "back" banks.

I have modified three of my existing programs to use extended memory instead of string and numeric arrays. This modification has allowed these programs to work with 5-10 times more data than before and frees up a large part of memory for expansion and enhancement of the program. The subroutines in the demonstration program that follows should provide a good start for anyone wishing to use extended memory from BASIC on a Model 4.

GETTING SET UP

The first thing to do to get started is to READ pages 12 and 13 of the LS-DOS 6.3 upgrade documentation. There is an inherent danger in fiddling with any operating system directly since there generally are no second chances, but if you understand what is happening, you can avoid inconvenience. If you aren't familiar with assembly language (like I'm not), just look at those "HL"'s and "DE"'s as places to store a byte of data, which is exactly what they are! The important thing is to remember what goes where and the sequence required to accomplish what you want. Sounds like BASIC, right?

Logical Systems has designed the BASIC SVC interface to use an integer array to hold the necessary data to pass to the SVC. The SVC returns data in these array elements to tell you what the result of the action was. This will be important later. For now, let's look at the setup. In the sample program, lines 1000-1070 are the heart of the interface. They load the control array AR% and transfer 256 bytes from extended memeory to a buffer at &H2300-&h23FF and vice versa. We have to tell it which direction to move data and from/to where.

This program segment is all that is needed to get or put 256 bytes into or out of extended memory. We must, however, find three numbers before we call the segment: PAGE%, the 256-byte "page" in the bank we wish to access, FUNC%, the EXMEM function we want to perform, and BANK%, the 32K bank number we want to access. These numbers are generated by the main program. I prefer to work with functions 3 (get 256 bytes) and 4 (put 256 bytes) and then extract the bytes I want. FUNC% is 3 for a function 3 operation (GET) and with BANK%=1 the value in AR%(2) is (3*256)+1 or 769. For a function 4 operation (PUT) FUNC% is 4 so AR%(2) is (4*256)+1 or 1025. If you convert these numbers to HEX, they become &H301 and &H401, respectively, which makes them easier to remember (the 3 or 4 for function and the 01 for bank number).

The PAGE% within the bank is the tricky part. One of the programs I wrote stores two integers X% and Y% and a single-precision number Z! for each "record" it generates. To put them in the buffer BUF$ use the disk-related comands MKI$ (for integers) and MKS$ (for single precision). The sequence becomes

```
BUF$=MKI$(X%)+MKI$(Y%)+MKS$(Z!)
```

MKI$ uses 2 bytes for each integer and MKS$ 4 bytes for each single-precision number. Here, we use 2+2+4 bytes to store the data. Since each record uses 8 bytes, we have to jump 8 bytes for the beginning of each "record" in the bank. There are 256 bytes in each bank access, so we get 32 of these records in each bank access. Confused yet? Just remember that when the access subroutine is called we will either put or get 32 8-byte records. BUF$ is used to hold each record as we work with it.

To get PAGE%, a simple calculation is required. Divide the record number wanted by the number of records per EXMEM access (32 in this case). The whole number left of the decimal point is the PAGE% and the decimal remainder multiplied by the number of bytes in the buffer (256) is the position in the buffer of REC%. For example, if we want record 550, we would get 550/32=17.1875. Then, 17 is the PAGE%. Once the buffer is filled, we can get the record back using a trick. The second and third bytes of a string descriptor are the location of the string in memory. We can change the memory location of the string by "pointing"it via two POKE statements. The sample program points the string to the 8 bytes in the record wanted. The decimal part of the above division (0.1875) multiplied by the number of bytes in the buffer (256) gives the offset in the buffer (48). So the 8 bytes in record REC% are at relative positions 48 through 57 in the buffer. Pointing BUF$ to this location provides an 8-byte "window" into the buffer.

Ok, now we can access the buffer, but it's an 8-byte string, not three numbers. How do we get it back? The MID$ function provides a "mask" to access the data returned from the bank in BUF$. We could have used straight integer and single-precision variables for this and avoided the number-string-number conversion, but using this method allows the access subroutine to be generic and we can mix and change variable types on the fly. This proved very important to me since I needed to keep the subroutines modular.

The subroutines at 200 (GET) and 400 (PUT) provide the logic to encode and decode the numbers. Using MKI$ and MKS$ changes the numbers to encoded strings, and CVI and CVS gets them back. We have to remember the location in each 8-byte string of our encoded numbers, but the program takes care of that. Lines 270 and 510 illustrate the method used to point BUF$ to the EXMEM buffer. POKEing the

VARPTR+2 of BUF$ sets the MSB byte BUF$'s location. A POKE to VARPTR+1 sets the LSB. The EXMEM buffer starts at &H2300 and ends at &H23FF, so the MSB is always &H23. The LSB is BUFPOS%, the offset in the buffer of REC%. Even though we use LSET in an attempt to keep BUF$ from being moved in memory by BASIC, it's always a good idea to get the VARPTR of a string before trying any manipulation.

The "INPUT" statements could be eliminated if the calling program already had established the values of X%,Y%,Z! and REC%. It should be getting obvious that there is duplication in the PUT and GET subroutines that could be placed into the EXMEM access subroutine, but if that was done, the EXMEM subroutine would be locked into using the data in the 8-byte chunks we discussed. By leaving the manipulation of BUF$ to the calling routine, we can change the data structure in the bank later. Let's say that we have 1000 of these 8-byte records stored in bank 1. That leaves 24,768 free in the bank that we could use for something else by simply adding an offset to PAGE%. Our 1000 8-byte records would use bytes 0 to 7999 (decimal) so adding 8000 to the calculated offset (PAGE%*256) would allow us to store other data above the 1000 records, using the same record numbers. I like to use this method to store labels and such that aren't used in calculations but are used to print reports. The record numbers, in effect, form the subscript of an single-dimension "array", and the offset becomes the key to access each "array".

AVOIDING TROUBLE

There is one very important thing that we haven't discussed so far: error checking. I like to assume that my computer is working properly, and since it usually tells me when it isn't (usually very rudely), I don't worry about checking for things like RAM integrity or malfunctioning hardware. Chances are that the malfunction would cause the program to crash before it could trap the error anyway. The one major error that can (and has) happened to EXMEM acess is forgetting to install EXMEM into the system. Since it has to be installed each time the computer is booted, it isn't out of the realm of possibility that it won't be there when needed. If EXMEM is SYSGENed, it will be in the system, but inactive. Roy Soltoff tells me that it must be installed each time, and it will NOT reuse its old memory slot if installed twice. DON'T SYSGEN EXMEM!. You will end up with two copies, one active, one not, occupying valuable low core RAM. That said, what happens if EXMEM is called and it is not present? The operating system presents an abrupt and unrecoverable ERROR X'2B' (SVC Error) and dumps you to LS-DOS Ready.

I added another USR11 routine to check for EXMEM in memory via the @GTMOD SVC #83 (lines 1100-1190). It searches memory for "$XM", the module name of EXMEM. The @GTMOD SVC sets the Z flag if it finds the module, and we can check the Z flag by ANDing 64 with AR%(0). If the result is a zero, the Z flag was not set and the module was not found. If you are interested in learning more about @GTMOD, see the Technical Reference Manual or Roy's The Programmer's Guide to LDOS/TRSDOS 6. I call the EXMEM finder right at the top of a program to avoid any lost time (like inputting 100 data items and discovering later in the program that EXMEM isn't installed). In my instance, all the programs used are called via single-key selection through a menu. I replace line 1180 with: 1180 IF A%=0 THEN SYSTEM"DO = EXMEM/JCL". The JCL then installs EXMEM and returns to the menu, so the operator can reselect the program. All this occurs in about 5 seconds (and very rarely), so the operator isn't inconvenienced. The JCL I use is:

```
.EXMEM/JCL
SYSTEM
EXMEM
BASIC MENU
//STOP
```

I have also added a line in my programs that tells what is happening and asks the operator to press any key to restart, so they aren't startled by the sudden return to the menu. There is one catch to this method of trapping the "EXMEM not installed" error. The @GTMOD SVC will quit searching if it finds a module without a name. Since all properly-written memory-resident modules have a module name, this shouldn't be a problem. If you have a resident filter, driver or program without a name in your system configuration, @GTMOD will quit and return a "not found" condition if it hits the offender before it finds EXMEM.

You must provide a means to avoid running past the end of the bank. The limit in my example would be 32768/8 or 4096 records. If more than one bank were to be used, the next bank could be selected when REC% exceeded 4095 (remember that REC%=0 is valid), but it will suffice here to put in an extra line to check for an overrange condition. The GET routine does this in line 225, in PUT it's 455.

Another error condition which can occur during a bank request is BANK IN USE. It is entirely possible to request a transfer to a bank being used by another program, like MEMDISK. Imagine the fun if we write 256 bytes into a RAM disk that contains the operating system overlays! The subroutine at 700-720 uses a USR11 call to the @BANK SVC #102 function 2 to tell if the bank we want is already in use. ANDing AR%(0) with 64 returns 0 if the bank is in use. I set AR%(1), AR%(2), AR%(4) and AR%(5) to zero to avoid any unpredictable results with this SVC, since an error could switch out the operating system or the BASIC program if the wrong function is called!.

Finally, to dress up the display screen a bit, I added another USR11 routine to scroll-protect the top two lines of the screen. Lines 600-630 enable scroll-protect to keep the title from scrolling off, and lines 650-670 disable scroll-protect. I used the SVC (@VDCTL #15) to accomplish this rather than

POKING the video DCB to avoid any incompatibility with future versions of LS-DOS. The demo program does not disable scroll-protect--you MUST type GOSUB 650 at BASIC's "Ready" prompt to disable the scroll-protection. I did it this way to avoid having to check for an exit conditon at each INPUT statement. To stop the program, just press <BREAK>.

Im sure that you have come up with many applications for extended memory by now, but for me, the novelty of running a 20K program using 32K of data in a BASIC work area of only 25K hasn't worn off. I can now use large "string arrays" and not have to deal with garbage collection or the massive waste of memory on descriptors, and my programs have not shown any noticible slowdown. I've changed the way I program to use arrays as pointers to extended memory data, so I can search, tag and sort data without having it all stored inside the program area, but that's a topic for another day. Happy Banking!

```
1 ' This is XMEM, a demonstration of extended memory
data storage using EXMEM
and USR11 for LS-DOS 6.3.  Written by Mark A.
Mueller, Compuserve 73100,2413.
2 ' Submitted for publication in The Misosys
Quarterly. Author assumes
absolutely no responsibility or liability whatsoever
for damage or inconvenience
3 'as a result of using this program or parts
thereof. In other words, have fun
but be careful! This program uses low-level
supervisor calls which can trash
your disks if used in error.
4 ' WARNING:  Before running this program for the
first time, open your disk
drive doors and write-protect any hard disk.  A typo
of an SVC number could
cause a FORMAT or other disaster.
10 CLS:DIM AR%(5)'        Create the integer array
for USR11
20 PRINT @ (0,20),"XMEM/BAS BASIC Extended Memory
Demo":PRINT STRING$(80,131)
25 GOSUB 600'  Scroll-protect title lines. Type
"GOSUB 650" at Ready to cancel
30 BUF$=STRING$(8,20)'    Initialize a buffer string
40 GOSUB 1100'            Check for EXMEM
50 PRINT:INPUT"Enter Bank # to work with";BANK%
55 IF BANK%=0 THEN PRINT"Can't use bank 0!!  Select
another.":GOTO 50
60 GOSUB 700:IF (AR%(0) AND 64)=0 THEN PRINT"Bank In
Use!! Select another.":GOTO 50'
Check for bank availability
100 PRINT:PRINT"Get data from a bank"
110 GOSUB 200'            Get Data from Bank
120 PRINT:PRINT"Put Data into a bank"
130 PRINT:GOSUB 400'      Put Data in Bank
140 GOTO 100
200 '             Subroutine to get data from a bank
210 INPUT"Enter Record Number desired (<ENTER> to
PUT)";A$
220 IF A$="" THEN 320 ELSE REC%=VAL(A$)
225 IF REC%>4095 THEN PRINT "Too High!":GOTO 210
230 PAGE%=INT(REC%/32)'  Calculate 256-byte page
which contains REC%
240 BUFPOS%=((REC%/32)-PAGE%)*256' BUFPOS%=position
of REC% in buffer
250 FUNC%=3'  GET function = 3
260 MSG$="...get page to buffer":GOSUB 1000
270 POKE VARPTR(BUF$)+2,&H23:POKE
VARPTR(BUF$)+1,BUFPOS%'  Point BUF$ to EXMEM's
buffer
280 X%=CVI(MID$(BUF$,1,2))'  Convert strings to
numbers
290 Y%=CVI(MID$(BUF$,3,2))
300 Z!=CVS(MID$(BUF$,5,4))
310 PRINT"Record #";REC%,"X%=";X%,"Y%=";Y%,"Z!=";Z!
320 RETURN
400 '             Subroutine to put data into a bank
410 INPUT"Enter an integer for X% (<ENTER> to
GET)";X$
420 IF X$="" THEN 550 ELSE X%=VAL(X$)
430 INPUT"Enter an integer for Y%";Y%
440 INPUT"Enter a number for Z!";Z!
450 INPUT"Enter a record number (0-4095)";REC%
455 IF REC%<0 OR REC%>4095 THEN 450
460 PAGE%=REC%\32'                    Calculate 256-
byte page # (0-127)
470 BUFPOS%=((REC%/32)-PAGE%)*256'  Calculate
position of rec% in page
480 FUNC%=3:MSG$="...get page to buffer"
490 GOSUB 1000'               Get page with record to
modify into buffer
500 FUNC%=4:PRINT"Loading data into buffer"
510 POKE VARPTR(BUF$)+2,&H23:POKE
VARPTR(BUF$)+1,BUFPOS%'  Point BUF$ at EXMEM buffer
520 LSET BUF$=MKI$(X%)+MKI$(Y%)+MKS$(Z!)' Load our
new data into EXMEM bufer
530 MSG$="...put buffer to page":GOSUB 1000
540 PRINT"Record #";REC%;"has been placed in bank
#";BANK%
550 RETURN
600 '                     Scroll-protect first 2 lines
610 AR%(0)=15:AR%(3)=&H702:X=USR11(VARPTR(AR%(0)))
620 PRINT"Type GOSUB 650 at 'Ready' to cancel
Scroll-Protect."
630 RETURN
650 '                     Un-scroll-protect
660 AR%(0)=15:AR%(3)=&H700:X=USR11(VARPTR(AR%(0)))
670 RETURN
700 '                     Bank Availability Test
710 AR%(0)=102:AR%(1)=0:AR%(2)=0:AR%(3)=&H200+BANK%:
AR%(4)=0:AR%(5)=0:X=USR11(VARPTR(AR%(0)))
720 RETURN
1000 '                    EXMEM Access Subroutine
1010 PRINT"Accessing bank";MSG$
1015 GOSUB 700:IF (AR%(0) AND 64)=0 THEN PRINT" Bank
in use! Transfer Aborted":GOTO 1070
1020 AR%(0)=108'          SVC number (108 is the
@EXMEM SVC)
1030 AR%(1)=PAGE%*256'    Offset within bank for HL
register
1040 AR%(2)=&H2300'       Tell EXMEM to use its own
buffer as our buffer
1050 AR%(3)=FUNC%*256+BANK%'    FUNC%=function #,
BANK%=bank # AR%(4) and AR%(5) are not used but
necessary for USR11 to work
1060 X=USR11(VARPTR(AR%(0)))'   Do it!
1070 RETURN
1100 ' EXMEM finder
1110 A$="$XM"+CHR$(3)'    The search name needs an
ETX at the end
1120 AR%(0)=83'           @GTMOD SVC
1130 AR%(2)=CVI(CHR$(PEEK(VARPTR(A$)+1))+
CHR$(PEEK(VARPTR(A$)+2)))'Tells @GTMOD where the
search name is in memory
1140 X=USR11(VARPTR(AR%(0)))'  Go find it
1150 A%=(AR%(0) AND 64)'  A%=0 if not found, A%<>0
```

```
if found
1160 IF A%=0 THEN PRINT"EXMEM Bank handler is not in
system"
1170 IF A%<>0 THEN PRINT"EXMEM is installed"
1180 IF A%=0 THEN PRINT"Program Terminated":END'
Put your recovery routine call here
1190 RETURN
2000 '     Bonus routine to examine EXMEM buffer in
8-byte chunks
2010 FOR X=&H2300 TO &H23FF
2020 Z=Z+1
2030 PRINT HEX$(X),
"&H";HEX$(ASC(CHR$(PEEK(X)))),ASC(CHR$(PEEK(X)))
2040 IF Z<8 THEN 2070
2050 PRINT (X-&H2307)/8;" --------------------------
--------":Z=0
2060 A$=INKEY$:IF A$="" THEN 2060
2070 NEXT X
2100 '     Bonus routine to fill an 8-byte buffer
record with a number
2110 BUF$=STRING$(8,32)
2120 INPUT"Enter fill integer (<ENTER> to view
buffer)";A$
2130 IF A$="" THEN 2010 ELSE A%=VAL(A$)
2140 INPUT"Enter buffer record # to fill (0-
31)";REC%
2150 IF REC%<0 OR REC%>31 THEN 2140
2160 POINTER%=REC%*8
2170 POKE VARPTR(BUF$)+2,&H23
2180 POKE VARPTR(BUF$)+1,POINTER%
2190 LSET BUF$=MKI$(A%)+MKI$(A%)+MKS$(A%)
2200 GOTO 2120
```

MISOSYS Products' Tidbits

DiskDISK

diskDISK [LDOS 5.x L-35-211] [DOS 6.x L-35-212]

Do you have a hard disk? If so, you need **diskDISK**. The diskDISK utility allows the creation of *logical disk partitions* as files on a physical disk drive. This is indispensable for hard disk users. Once a **diskDISK** file is *installed* into a logical drive slot, the diskDISK can be used just like any other physical drive; **diskDISK** provides for easy swapping of any currently active diskDISK file.

With **diskDISK**, you can easily group related files for ease of maintenance. DiskDISK files can also be set up as *images* of physical drives to allow mirror image backups.

Finally, diskDISK drives allocate in granule sizes smaller than your hard disk system. Five inch diskDISK images allocate just like floppy drives. Also, there are special diskDISK types that allocate in one or two sector granules for maximum storage efficiency.

Large diskDISKs

Fm Dave Krebs: I just got DiskDISK the past couple of weeks (model 4) and installed a couple of 180k 5" disks on my 5 meg R/S drive (RSHARD drivers with 1 meg gran size). Right now I'm having no luck installing a 360K double sided 5" drive. Too many extents (> 12) is the error. My single sided drive has 7 extents. The platter that I'm trying to install the 360K drive on has almost 500k of contiguous space according to the FREE command, any suggestions?

Fm Pete Granzeau: Dave, I had a problem like that, and using MAPPER (from the Mark IV collection) found that the free space was terribly checkerboarded (lots of one or two-gran holes in the map). My solution was to CREATE a dummy file big enough to put all my DiskDISK files out at the end of the HD, following the DIR/SYS file. You can then REMOVE the dummy file.

Fm Dave Krebs: Pete, I don't think that's my problem, I did have a couple of ones like that on one drive but the other was clean (according to FREE). I think I'll look up that memory command to change the value of AFLAG$ maybe that will help!

Fm Kerry Wilson: Dave, Are there a lot of "small holes" before you get to the contiguous area? If so, you might try what I normally do with my 360K diskDISK files. Use the memory command to set the AFlag to a high enough value so when you issue the DDFORM command, the file will be written to the cylinder you want.

Example: Let's say your contiguous area doesn't start until cylinder 80. Do the following BEFORE creating the diskDISK file with DDFORM.

```
MEMORY (A="A",B=80)
```

What this does (in case you're wondering) is tell the disk operating system to start with track 80 when looking for room to create a new file. Don't forget to set it back to its normal value after you've finished creating the new files you want. [i.e. `MEMORY (A="A",B=1)`]

Fm Dave Krebs: Ah Ha, Kerry, that was the memory command that I was looking for. I had it stuck somewhere when I was making a double sided boot disk. This time I'll file it in my Manual in the right place.

Fm Joe Kyle-DiPietropaolo: Dave, The problem is the 1K gran size. A fact that may people don't know is that in all the LDOS/TRSDOS 6 and other systems that use the original Randy Cook directory structure (this leaves out Model 3 TRSDOS 1.3), an extent is defined as a block of contiguous granules up to 32. IF there are more than 32 grans needed for a file, another extent must be allocated even if the space happens to be contiguous. This is a byproduct of the way that the extents are encoded into fields within the FDPE and FXDE entries.

Generally, if you are going to use diskDISK you pretty much dedicate logical drives to holding diskDISK files, format those drives with larger granules and the problem will go away.

Fm Roy Soltoff: Dave, diskDISK doesn't necessarily look for the largest contiguous space on the host drive since it's using the DOS to allocate the space. Suggest you alter the starting cylinder (one of the FLAGS) to point to that 500K

region just before the DDFORM, then set it back to 1 afterwards.

Fm Dave Krebs: Roy, Later I thought of that, but couldn't find the right command. Looked around in '*Guide to TRSDOS 6*' and found the right FLAG$, and the right memory command came from another message in this string of 5. Thanks!

Aerocomp DiskDISK FIX

Fm Jim Beard: Roy, I scarfed down your new upload [see *Help with Aerocomp drives*, TMQ II.iii, pg 56] and tried it on my DiskDISK. My PATCH utility kept reporting a find line error, but a hex listing said there wasn't a problem. So, I used FED to patch my DD/CMD. It works great with my AEROCOMP 20 Meg HD with Adaptec drivers. At last, I have my user-controlled subdirectories on LS-DOS. Many, many thanks.

Fm MISOSYS, Inc: Strange about that error report. Wonder if there was a "weird" character in the file.

Fm Jim Beard: Roy, There WAS a weird character in the patch file. I forgot to comment out the file name, which just happened to begin with "d". Once I did that, the patch worked fine.

More on Aerocomp DDFIX

Fm Jim Beard: Roy, I have discovered a strange idiosyncrasy to the DiskDISK patch for the Aerocomp drivers for Adaptec controllers. My configuration is a Seagate 225 (20 Meg) operated as drives 0-3, with floppies as drives 4-7. I use DD to replace drives 6 and 7. My DD files are on :2, and I find that I can't make a DD file on :3 work. Somehow it either doesn't come up with a viable "boot track", or it is a clone of a DD file on :2. Multiple files on :2 are OK. There is nothing in high memory except the Aerocomp driver.

I don't have a problem, so just file this info away for reference. On the other hand, if you have any insight, please share it with me.

Fm MISOSYS, Inc: That's too strange to even conjecture a reason.

Fm Jim Beard: Roy, I tried removing the file and regenerating it several times, and had no luck until I dropped back to :2. Then, no problem. Can you have several DD's active at once? Once I copied my good DD from :2 to :3, I got a copy of it every time I tried to DDFORM :3, and the files would list and everything. I deleted the old file, so DDFORM probably opened up the new sectors. It's as if DDFORM simply doesn't write the boot track on HD :3.

Fm MISOSYS, Inc: That would be kind of impossible. DDFORM is accessing the disk via the standard DOS file access calls since it is referencing a FILE. It could care less if that file is on drive 0, 1, or whatever. Now I have my suspicions that maybe your driver is funny. But without the time or the equipment to test that out in detail, I couldn't say what the cause is.

Incidentally, you can have as many active diskDISKs as you have free drive slots.

Fm Jim Beard: Roy, To say my driver is funny is like saying the Pope is Polish, and that all his predecessors are Italians. It is an EARLY Aerocomp driver for Adaptec controllers on Seagate 225 HD's. I think the real answer is to get new drivers from Aerocomp, which they have graciously offered to send me for free, AND keep the patch on DD. Thanks for your interest and response. And, THANKS FOR THE PATCH, as I can use DiskDISK now as is.

EnhComp BASIC compiler

EnhComp [DOS 6 M-21-072] [LDOS 5 M-20-072]

This is an enhanced **BASIC compiler** released in 1986 and reviewed in the March 1987 issue of *80 Microcomputing* and October 1987 issue of *COMPUTER SHOPPER*. This compiler has lots of great features. It handles the bulk of Model III Microsoft BASIC and supports additional commands and functions. Standard is floating point with both **single and double precision** functions; random file access ("X" mode for reclens to 32767), turtle graphics, pixel graphics, keyed array sort, multi-lined functions, user commands, IF-THEN-ELSE, REPEAT-UNTIL, printer control, sequential file positioning, line labels and more.

A *supervisor* program automates the edit-compile-test phases inherent when using compilers; this makes using EnhComp almost as easy to use as your BASIC interpreter. You also get CED, a **line editor** with string search/change, partial load/save, renumber, copy, and move.

Enhcomp has a built-in **Z80 assembler.** You can **easily** create hybrid programs of BASIC statements and in-line assembly code which **completely eliminate** contorted string packing and DATA statement high-memory module techniques for your BASIC program to access a machine code module. Z80-MODE accesses BASIC's variables!

You'll have to edit *existing* BASIC programs, but the power and completeness of EnhComp make that an easy task.

TMQ/Enhcomp

Fm Ken Stiefel: Roy, Roy Having finally gotten around to reading the Winter TMQ (not from lack of interest, just lack of time I discovered the little not about OUT not working in Enhcomp. Thank you for printing that. That explains why a

control program worked from the interpreter, but played very dead after it was compiled. Please add me to you list of people waiting for the fixed version of SUPPORT/DAT. I do have one question though. Did people really call up wanting the upgrade asking for the WHITE Computer. I didn't think people came that dumb. Thanks again for turning out some excellent products, and please don't get so frustrated dealing with the screwballs of the world that you give up on the rest of us.

Fm Roy Soltoff: Ken, Just send back your EnhComp disk. There's a lot of other things we fixed up. I have no bugs pending with EnhComp other than a very minor one: don't try to put two separate DIM statements into a program. EnhComp can have only one of them. I am contemplating a patch to trap that problem.

Fix for weird problems

Fm MISOSYS, Inc: We have recently uncovered a problem with some BASIC programs ported to EnhComp which then blow up in unpredictable ways. We have isolated the problem to the following. Interpretive BASIC does not require you to DIMension arrays unless the single dimension exceeds 10 (i.e BASIC will default to an array size of 10). EnhComp, on the other hand, requires that all arrays be dimensioned. Unfortunately, the requirement, although it had a run-time trap, didn't have a total fail-safe trap. EnhComp generated a run-time error when you tried to reference an array element when the array had not been DIM'd.

Unfortunately, if you had an undimensioned STRING array and the program executed various string manipulations before that undimensioned string was referenced, a need to shift string space would crash the program. We have developed a patch to the compiler to trap and inform of undimensioned arrays. In the interim, please ensure that you have dimensioned all arrays with an EXPLICIT DIM statement.

EnhComp Bugs & Suggestions

Fm J. N. Grindey: I have been having a few problems with PRO-EnhComp. I wondered if the disk may have been slightly damaged (magnetically) in transit as I don't think I would be the first to have found these problems but try them for yourself.

This example programme will save the data to disk all right but when it tries to read it back, it prints to 88 (the end of the first sector) then gives runtime error code 124 end of file encountered. If the semicolon at the end of line 40 is removed it then works O.K.

```
10 ALLOCATE 1
20 OPEN "O",1,'"TEST/DAT"
30 FOR I = 1 TO 150
40 PRINT# 1,I;
50 NEXT I
60 CLOSE
70 OPEN "I",1,"TEST/DAT"
80 FOR I = 1 TO 150
90 INPUT# 1,A
100 PRINT A
110 NEXT I
120 CLOSE
```

This example programme will ignore the last CHR$() in line 40:

```
10 CLS
20 A$=WINKEY$
30 PRINT "A$=";ASC(A$)
40 IF
INSTR(CHR$(8)+CHR$(9)+CHR$(10)
+CHR$(13),A$) THEN PRINT
ASC(A$)
50 GOTO 20
```

Print USING: while using the format "**####.##-**" if the number is 0 to the 2nd decimal place (i.e. 0.00345), that is what gets printed not 0.00. Also with the same format the "-" works as in the manual but would it not be better to work as in, Microsoft Basic and print a trailing space if the number is positive instead of nothing so that anything printed twice in the same position the first time a negative number the second positive, then the trailing "-" does not get blanked out so the number still looks negative; (i.e.) suggested format:

```
USING "####.##.-";15.69 =" 15,96 "
USING "####.##-";-15.96 =" 15.96-"
```

I bought a compiler because I thought they were quicker than normal basic but the following loop take twice as long using PRO-EnhComp than MODEL 4 BASIC is this normal?

```
10 CLS
20 PRINT TIME$
30 FOR I = 1 TO 10000 : NEXT I
40 PRINT TIME$
```

Are the following allowed? **OPEN "O",1,"*WL"** or any other device driver? This gives a runtime error 178 (attempt to

open file with different length). `FIELD 1,6 AS FH$(1),6 AS FH$(2)`, etc. It doesn't field string arrays.

On page 3-7 of the manual [referencing] the "Q" command, it states that if drive number is omitted, drive 0 is assumed; Q <ENTER> returns with a parameter error.

Totally different subject, ALTDISK from PRO-ESP. While modifying it to use up to 256K, I noticed that the GAT media block GAT+X'F5' TO CAT+'X'FF' is actually placed at GAT+X'F4' TO GAT+'FE'. This doesn't affect the program as far as I can detect, can it?

Fm MISOSYS, Inc: J. N., Because of a few bug reports which have been received in the past few months, I have delayed in resolving them individually. The reason behind that is that some problems can only be fixed by making coding changes and then re-assembling the affected modules. I have upgraded EnhComp to version 2.6 which corrects the problems you experienced as well as others. A copy of the release has been placed onto the enclosed diskette which is being provided to you. As all modules were altered, please use this as your new "master" disk. The README/TXT file has been totally revised; please read it in its entirety. Sorry for the delay, but too many things got into the way here and revising the EnhComp package to release 2.6 was a significant undertaking. We also had to spend the time to prepare *THE MISOSYS QUARTERLY* issue II.iii for publication.

Specifically, <1> was fixed with a patch appearing in TMQ II.iii which involved the INPUT# statement. <2> is fixed in the 2.6 release. I have revised the USING formatting to conform to Microsoft BASIC in the use of the trailing minus sign. Thanks for the suggestion.

As far as the slowness of the EnhComp-compiled sieve (your point <4>), I think it has to do with the fact that EnhComp compiles not to native code but rather to a form of pseudo-code. Also, unless you compile with the -NX option, the trace code and break handling code gets compiled into the program; this executes the break testing and trace testing code on every statement execution. That by itself may add 5-10% in execution time. I tested your program on one Model 4 with the following times: EnhComp - 17s. EnhComp with -NX - 16s, M/S BASIC - 11s. One of the things most folks don't realize about M/S BASIC on the Model 4 is that once it interprets a statement involving a scalar variable, it resolves the address of that scalar; thus, for iterative loops, it doesn't waste the time. It may be interesting to note that the times on a Model 4 running in Model III (FAST) mode are 18, 17, 14 respectively. On the other hand, the true test of a compiler operation is more than just evaluating the speed of an empty loop. EnhComp gains you many features - in-line Z80 code, graphics, block structured IF statement, user defined commands, multi-line functions, etc., to name a few.

I have modified the OPEN handler to permit opening devices as noted in your point <5>. Also, both the compiler and the support library were modified to fix the problem of string array elements used in a FIELD statement.

Finally, the "Q" command was fixed in the model 4 version of CED to default to drive 0 if no drive number was entered; it was correct in the Model III version.

We discontinued PRO-ESP early last year. We just don't have the time to spend on it. The problem you address is minor; it should cause no un-manageable problems.

EnhComp 2.6 release

Fm MISOSYS, Inc: Because of several bug reports which required re-coding and re-assembly to correct, I have revised the EnhComp (Mod 3 and Mod 4) packages to 2.6 release. Fixing up the problem with using string array elements in a FIELD statement caused the biggest debugging problem, but I now have that working.

I also added code to BC/CMD to trap undimensioned arrays. After the first pass, a routine goes through loops based on the number of total variables used. It then isolates non-string scalers and then string arrays. Since an undimensioned array is still a variable, the array control block is allocated. But SSR42 which executes the DIM statement is the runtime routine which loads up the control block. If there is no DIM, then kablooey on string arrays if the string space needs adjustment. A long time ago I did indeed put a trap into the DIM routine to generate a run-time error out of SSR41 (which resolves an array varptr) if the array had not been dimensioned. That was fine for everything but strings. Now the compiler trap will not permit compilation - it as a terminal error after the first pass; it also reports the variable name in question.

I also fixed up trapping record lengths greater than 255 for non-X modes in the OPEN statement. I also added the ability to field a "R" mode file with a 256-byte string; that was previously unusable. I also added some code to trap a field overflow; that was not done previously. The following summarizes the changes:

1. Compiler now traps undimensioned arrays;

2. POS and ROW are fixed in M4 version;

3. Trailing minus on USING spec now works like Microsoft BASIC;

4. RMARGIN=0 now inhibits all paging control on lprint output; useful when sending graphics output;

5. OPEN provides error handling on reclen>255 for R-mode files;

6. FIELD now supports a single field of 256;

7. FIELD now supports string array elements;

8. REF command fixed for Model I users;

9. Possible PHASE error on END statement usage eliminated;

10. TYPE() function syntax error corrected;

11. RESUME line# can now resume from any level (GOSUB routine, FOR-NEXT loop, etc) where line# is at highest level (not in a loop or subroutine); [read that, Bryan!]

12 CED (mod 4) Q command fixed for default to reference drive 0;

13. A bug in INSTR() using a concatenated string expression has been fixed;

14. OPEN has been improved to permit opening of devices;

15. FIELD now traps field overflow if the total fields exceed reclen;

16. LINEINPUT (mod 4) now turns on the cursor during the input.

To get the 2.6 release, send in your EnhComp master disk and supply a return address label. There is no charge for this release update.

EnhComp and OUT

Fm Mike Harrow: [10-Apr-88] Roy, Just had a chance to read all the new messages. You received a note from Ken Stiefel about EnhComp and the OUT command. Your reply to him leads me to believe that you've recompiled the SUPPORT/DAT and corrected this command.

Version 2.6 is in my hands and the OUT command still won't function. This leaves me rather perplexed. You mentioned the correction would require a recompile of SUPPORT/DAT. I hope you didn't go to all that trouble then forget "OUT".

Fm Roy Soltoff: Mike, I dug into the EnhComp problem with OUT which you referred to me the other day. I did have corrected code in the 2.6 release; unfortunately, I corrected the code to support the statement, OUT VALUE,PORT. The correct syntax for OUT is OUT PORT,VALUE. I had to revise the code. Again, unfortunately, it required one additional byte so I couldn't just make it a patch.

Note to those few receiving the "early" EnhComp 1.6 update. If your README/TXT file does not have the note about the OUT correction referenced here, please return your disk one more time. Sorry for the inconvenience.

Floating point woes

Fm Michael Dauphin: Roy, I am having a small problem with Pro-EnhComp. Try this program:

```
PRINT 2^4
X% = 2^4
Y! = 2^4
Z# = 2^4
PRINT X%;Y!;Z#
```

result: 16 15 16 15.99999999999607

This 'problem' was giving me fits when I was trying to set bits in a byte. Am I doing something wrong or could this possibly be a bug. I just received EnhComp (less than a week after I ordered it!) so maybe I missed something in the manual. It just seems to me that X% = 2^4 SHOULD return 16.

Fm MISOSYS, Inc: I suspect that inexactness of floating point enters into the fray. For a good down to earth article on this topic, see page 217 of the March 1988 issue of *BYTE* magazine. The article is "*Floating Point Survival Kit*".

Most folks don't understand the inaccuracies associated with representing numbers in binary floating point. Your PRINT statement would have to reference a USING which forced round off to whole numbers.

Fm Joe Kyle-DiPietropaolo: Michael, I'll defer to Roy's expert information, but it looks to be like the power function is implemented using floating point approximations, thus the result when using ints. I'd bet that Z# = 2*2*2*2 works ok.

Fm MISOSYS, Inc: Power functions in BASIC are invariably implemented using the lof function. Or actually the exp function.

Fm Joe Kyle-DiPietropaolo: Roy, Calculate X to the Y power using the lof function? Sounds very, very slow... HaHaHa! Anyway, 2*2*2*2 should work exactly, but puts the calculation overhead at runtime, unless EnhComp is better than I think.

Fm MISOSYS, Inc: A few too many blibs got in the way of the typing. Actually, the exp function is what's used. Let x**y = e**z. Then take natural logs of both sides. You get y ln x = z ln e = z. So evaluate y ln x then use the exp function.

Fm Phil Oliver: Brief note, Joe: "x^y" is being calculated using the identity: e^(y * ln(x)), inside ENHCOMP. I ought to know ...

Fm Joe Kyle-DiPietropaolo: Phil, Sounds like a good plan to me. Treating integer arguments as a special case would

be messy. Too bad many folks don't fundamentally understand the drawbacks (and advantages) as to how most computers deal with floating point math and transcendental functions...

Fm Shane Dawalt: Roy, When you said that floating point (and inaccuracies generated therein) would be over the heads of most TMQ readers ... I agree. It is quite a nasty little (little? I mean HUGE) concept to grasp. Using my own words ... a bloody bear of a problem.

Fm MISOSYS, Inc: But to tell the truth, anybody who does any programming using floating point must, repeat MUST, understand the underlying concepts and inaccuracies associated with floating point calculations by a computer. Give us back the 3-significant digit slide rule!

Fm Michael Dauphin: Roy, The short sample I [previously] included produced the output just as shown. The PRINT statements did NOT reference USING. Any rounding was performed by EnhComp. Here is another example:

```
Y! = 4
X! = 2
A! = X!^Y!
PRINT A!,BIN$(A!)
```

Result: 16 0001111

It looks to me as if 4!^2! does not return 16. Just a close approximation (15.99999...). Then EnhComp's PRINT statement is rounding so it seems as if the correct answer is returned by EnhComp. I tried 2%^4% with Mod III basic. It does return the right answer. All of this is no big deal. Now that I am aware of the 'problem' I can work around it. EnhComp's excellent FUNCTION statement makes it an easy task to write a short integer routine which calculates the values I need without any messy code in the main body of the program [exp% = !EXPO(base%,exponent%)]

By the way, I would have bought EnhComp when it was first announced a couple of years ago if I had known that it allowed long variable names. It was not mentioned, at least I didn't see it, in any of your lit. I don't even remember seeing it mentioned in the Computer Shopper review. I learned a long time ago that with most software: 'if it don't say; it ain't got it.' It wasn't until I called Misosys a couple of weeks ago and asked that I found out that it does indeed support long names.

One more thing, Any chance something like the Mark IV collection will be available for 6.3? I bought the Mark III collection for my mod I and think it is the best $100 I ever spent!

Fm Roy Soltoff: Michael, Most folks dealing with computers today think they are accurate. They are not. When you use floating point arithmetic, you introduce inaccuracies because floating point cannot accurately resolve all numbers. Although you think that you are dealing with whole numbers when you want to raise 2 to the 4th power, think how the computer has to compute that. It has to take 4 times the natural log of 2; then it has to exponentiate that. This introduces two functions: exp(x) and log(x). Both of these functions require either a truncated Taylor series or a Chebychev series (again truncated). These series use definitely non-integral numbers. They are only approximations. Albeit, good approximations when you only use no more than the significant digits you started with. The problem comes in when you start off with a number which has one significant digit and use inexact floating point (single precision has 6-7 digits of precision). If you compound the problem further by using double precision (14-16 digits of precision), you can then, when you introduce log() and exp() functions, experience an in-accurate result with 5-14 trailing decimal places.

Is a number 15.999999999847521 less accurate than 16? Absolutely not when you round it off to contain no more digits of significance than what you should be permitted to accept. Perhaps its time for a good article on the subject to appear in TMQ; however, I think that the subject matter would be over the heads of most readers.

The MARK IV collection covered the Model 4 but that collection has been sold out. The programs making up the Collection were poor sellers; that's why they weren't developed further for x.3 directory support.

REF bug and benchmarks

Fm Larry L Hildebrand: Hello Roy and family. Congratulations to you and Brenda on your expected new addition to the family. With two girls you should be thinking blue and giving Brenda many gifts all in blue. My wife, Paula Rae, and I wish Brenda good health, an un-eventful delivery, and a healthy addition to your family.

I look forward to receiving each issue of *THE MISOSYS QUARTERLY*. I especially enjoy the section about your family and what is going on. As we are in our early 50s and our two kids are grown, daughter 29 and son 25, I get a chuckle and grin when I read about Stacey and Stefanie growing up. We had the swing set, sliding board, and wading pool situation here with a back yard full of kids also. Enjoy your kids when they are young; they will grow up to soon.

Roy, is UNREL for TRS80 for a model I? In the catalog 87-1, page 21, last paragraph states UNREL includes programs for both the model 4 and model I/III. However, the next line Ordering Information just lists models III/4.

I am having a problem with the REF/CMD utility program that came with the EnhComp Basic Compiler. The sign on message is garbled and I get the same parameter error every time. Maybe the master disk got zapped somewhere between there and here? My master disk is enclosed. Could you refresh

the (REF/CMD) reference program and return? If there is a charge let me know.

I am enclosing a table of running times for the Eratosthenes Sieve Prime-Number program (80-Micro Dec 1985 pages 48-50). The Pascal program was compiled by Pascal-80, C program by your LC program, the basic compiled by your EnhComp basic and ZBASIC.

	Execution Time (sec)
Interpreter Basic TRS80 Model I	3508
Basic EnhComp Compiler	2312
Basic ZBASIC Compiler	320
Pascal Pascal80 Compiler	1608
'C' LC Compiler	587

Alcor Multi-Basic compiler would not compile the program due to insufficient memory.

The execution times were what I expected, except for the ZBASIC program. I had figured that the "C" program would have produced the fastest runtime followed be the Pascal program, Compiler basic and interpreter basic, which was not the case. Don't get me wrong, I am not criticizing "LC", "Pascal-80" or "EnhComp", nor am I trying to put a feather in the cap of ZBASIC. The question I can't answer is "WHY?"

I am enclosing a Model I, DD/40 track disk with programs on it. No need to return the disk. If you, or someone who has the time and expertise, have an insight as to why the drastic runtime differences, I would appreciate it.

This maybe some fodder for an article in your quarterly. Thank you Roy, for the excellent quarterly and even better software.

Oh, by the way, if you are interested I work for The National Weather Service here in Chicago at O'hare International Airport as an Meteorological Technician. Quite a busy place.

```
00010 REM sieve/bas * as run Under Zbasic Compiler
00020 A$=TIME$:PRINTA$
00030 DEFINT R-Z
00040 DIM FLAGS(8191)
00050 PRINT "10 ITERATIONS :"
00060 FOR I = 1 TO 10
00070 PRINT I
00080 COUNT = 0
00090 FOR J = 0 TO 8190
00100 FLAGS(J) = -1
00110 NEXT J
00120 FOR J = 0 TO 8190
00130 IF NOT FLAGS(J) THEN GOTO 220
00140 PRIME = J + J + 3
00150 PRINT USING"########";PRIME;
00160 L = J + PRIME
00170 IF L => 8190 THEN GOTO 210
00180 FOR K = L TO 8190 STEP PRIME
00190 FLAGS(K) = 0
00200 NEXT K
00210 COUNT = COUNT + 1
00220 NEXT J
00230 NEXT I
00240 PRINT COUNT;" PRIMES."
00250 A$=TIME$;PRINT A$
```

```
100 'SIEVE/BAS * AS RUN UNDER ENHCOMP Compiler
110 DEFINT A-Z
120 DIM FLAGS(20000)
130 PRINT "10 ITERATIONS: "
140 FOR I = 1 TO 10
150 PRINT I
160 COUNT = 0
170 FOR J = 0 TO 8190
180 FLAGS(J) = -1
190 NEXT J
200 FOR J = 0 TO 8190
210 IF NOT FLAGS(J) THEN GOTO 300
220 PRIME = J + J + 3
230 PRINT USING"########";PRIME;
240 L = J + PRIME
250 IF L => 8190 THEN GOTO 290
260 FOR K = L TO 8190 STEP PRIME
270 FLAGS(K) = 0
280 NEXT K
290 COUNT = COUNT + 1
300 NEXT J
310 NEXT I
320 PRINT COUNT;" PRIMES."
330 PRINT TIME$
```

```
/* Benchmark SIEVE/CCC */
/* compute primes using the Sieve of Eratosthenes */
#include stdio/csh
#define SIZE 8190 /* size of the number array */
#define SIZE1 8191 /* SIZE + i */
#define NTIMES 10 /* # of time to execute loop */
char flag[SIZE1];
main()
{
    int i, j, k, count, prime;
    printf("%d iterations: ",NTIMES);
    for (i = 1; i <= NTIMES; i++)
    {
        count = 0,
        for (j = 0; j <= SIZE; j++)
        flag[j] = TRUE;
        for (j = 0; j <= SIZE; j++)
        {
            if (flag[j])
            {
                prime = j + j + 3;
                printf("%8d",prime);
                for(k=j+prime;k<=SIZE;k+=prime)
                    flag[k] = FALSE;
                count++;
            }
        }
    }
    printf("%d primes.\n",count);
    exit(0);
}
```

```
PROGRAM SIEVE(OUTPUT);
CONST
SIZE = 8190;
SIZE1 = 8191;

VAR
I, PRIME, K, COUNT, ITER : INTEGER;
FLAGS : ARRAY[0..SIZE1] OF BOOLEAN;

     BEGIN
     WRITE('10 ITERATIONS');
     FOR ITER := 1 TO 10 DO
          BEGIN
          WRITELN(ITER);
          COUNT := 0;
          FOR I := 0 TO SIZE DO
          FLAGS[I] := TRUE;
          FOR I := 0 TO SIZE DO
          IF FLAGS[I] THEN
               BEGIN
               PRIME := I + I + 3;
               WRITE(PRIME:8);
               K := I + PRIME;
               WHILE (K <= SIZE) DO
                    BEGIN
                    FLAGS[K] := FALSE;
                    K := K + PRIME;
                    END;
               COUNT := COUNT +1;
               END;
          END;
     WRITELN(COUNT,' PRIMES.') ;
end.
```

```
5 ' Sieve/bas * as run under basic interpreter *
10 A$=TIME$:DEFINTA-Z:DIMFL(8191):
     PRINT"10 ITERATIONS: ":FORI=1TO10:
     PRINTI:CO=0:FORJ=0TO819O:FL(J)=-1:
     NEXTJ:FORJ=0TO8190:IFNOTFL(J)THEN50
20 PR=J+J+3:PRINTUSING"########";PR;:
     L=J+PR:IFL=>8190THEN40
30 FORK=LTO8190STEPPR:FL(K)=0:NEXTK
40 CO=CO+1
50 NEXTJ:NEXTI:PRINTCO;"
PRIMES.":B$=TIME$:PRINTA$:PRINTB$
```

Fm MISOSYS, Inc: Larry, I have upgraded EnhComp to version 2.6 which corrects the problem you experienced (REF command) as well as others. A copy of the release has been placed onto your diskette which is being returned to you. All patches which have been applied to release 2.5 have been incorporated into this 2.6 release (I hope). As all modules were altered, please use this as your new "master" disk. The README/TXT file has been totally revised; please read it in its entirety.

Sorry for the delay, but too many things got into the way here and revising the EnhComp package to release 2.6 was a significant undertaking. We also had to spend the time to prepare *THE MISOSYS QUARTERLY* issue II.iii for publication. Speaking of TMQ, it would have been more helpful if you had included the source code to the various Sieve programs rather than just the listings on paper; your diskette contained only the executable files. Nevertheless, it's food for a future TMQ. [Now that we have our scanner and OCR software, it wasn't necessary to re-type your source code; although the OCR software is not supposed to handle dot matrix except for NLQ fonts, it did a credible job on your "purple" listings. Some minor editing completed the job.]

As far as the slowness of the EnhComp-compiled sieve, I think it has to do with the fact that EnhComp compiles not to native code but rather to a form of pseudo-code. Also, unless you compile with the -NX option, the trace code and break handling code gets compiled into the program; this executes the break testing and trace testing code on every statement execution. That by itself may add 5-10% in execution time.

Questions, and other tidbits

Fm Steven C. Jerkins: I wanted to jump for joy when I saw the patch to DD/CMD in TMQ II.ii. I have been trying to get DiskDISK to work on my Aerocomp Hard drive for over a year now. Right now I am waiting for the hard drive to return from repairs. I don't know if it took a lightning hit or was damaged in shipping but when I finally got the household goods up to Norfolk from Charleston, SC the hard drive was not working. The bubble had to be replaced.

A bit of information that TMQ readers might be interested in. Montezuma Micro has written a new driver for the 20meg Aerocomp drive that they say is specifically for LS-DOS 6.3. The do not consider it an upgrade of the TRSDOS 6.2 driver so they charged me the $30.00 for a new driver. I'll send a report of its performance via CIS when I get it working.

The transfer from Charleston to Norfolk got my mail all bollixed up. I received TMQ II.ii a week ahead of TMQ II.iii. I wanted to strangle the Navy postal-system when I saw the, already expired special on DiskDISK. I still need to get the LDOS version of DiskDISK but have been holding off until I was assured that LS-DiskDISK could work on my hard drive. Phoooeh!!!!!!!!

By the way, can you get back issues of Disknotes? I lost the chance to send in the coupon with a couple of issues due to the flailex of a military transfer and a period at sea.

A while ago, when Tandy did the megadump of their Model 4 software, I got a copy of Alcor "C". Try as I might, I can't even get the programs I type in out of the manual to work. Alcor will not even answer their mail or telephone questions about TRS-80 products any more. They can't even tell me if the package I purchased is the latest version or not. As soon as I can get the cash I want to trade it in for your "C" package. The question is, how long is the trade in offer good for? Do I need to go hock something to get it now or will the offer be open in a month or so.

I'm interested in a compiled BASIC also. I have decided on either Pro-EnhComp or Zedcor, Inc.'s Zbasic. I looked over some of your advertisements for Pro-Enhcomp and the articles in TMQ and have a few questions to clarify some points. Does a file from a Pro-EnhComp compiled program stand-alone? It seems from your literature that it is NOT like BASCOM and requires some file to be resident on the system for it to run.

Secondly, can Pro-Enhcomp support the MicroLabs High Resolution Graphics Board? If it does not directly, how hard is it to interface machine language for controlling the HiRes board? Zbasic directly will support the board but I find you have to imbed Z80 code into the program to fully utilize some of the hardware.

Do you know of a patch to Pads or PRO-Pads to change the internal dating scheme? I realize that you had to discontinue this very good program due to lack of interest. Thinking about it, would it require a complete re-write of the program to change the date handling? It just looks tacky to have all the files I entered into a PaDS this year dated 1980.

Before I joined the Navy I worked at a Tandy Computer Center. I well understand the frustration with the "User" vice the "computer literate". I got disgusted with Radio Shack when it became evident that the whole computer sales system was being populated with "user" types and you were labeled as "weird" if you actually understood what you were talking about. I once had a customer with a Model 12 come in to get the computer fixed after he put a Model III copy of Zork into the 8 inch drive to play the game.... sigh!

I am considering getting the XLR8er board for my Model 4. Let me describe my system and see if you see any blatant hardware conflicts right off the bat. Gate array Model 4 (1069A model) with 128K memory, Microlabs Hi-Res board, Aerocomp 20meg hard disk, Orchestra 90 unit, 4 floppy drives (1 R/S standard, one DSDD from an IBM XT, and two external Mitsubishi 1.2 meg drives like the Tandy 2000 used). I read over the installation instructions in the latest TMQ and it does not seem as difficult as I had thought. Reading the instructions for the AlphaTech megamem board scared me off of that one. I would not ever be able to get an AlphaTech board out if I ever had to take the Model 4 in to Tandy for service.

A final note. I like the laser printer fonts. I thought when I saw the skinny size of the latest issue that you were starting to skimp. NNML! (navyese - Nay-Nay-Moose-Lips). I did a little math and with the change in fonts you are packing as much if not more information into a smaller package. The quality is excellent! Keep up the good work.

Fm MISOSYS, Inc: Steven, the 50% off on trade-in of equivalent non-MISOSYS products is an on-going offer; currently, there is no expiration of that offer. If we decide to eliminate the offer, our customers will receive adequate advance notification. Also all back issues of Disk Notes are available at $10 + S&H.

Now on to PRO-EnhComp. Once a source program is compiled, it is a stand-alone program; it needs no other program in order for it to run. EnhComp does not support the MicroLabs hires graphics (nor Tandy's, either). If you know assembly language and know how to program the board, you can easily interface the board within a BASIC program. EnhComp **permits inline assembly code!** Thus, you can program directly in assembly language. Usually this is done in a subroutine.

There is no patch to PaDS to change the internal dating. Only three bits were allocated for the year. PaDS is also a discontinued program because it sold fewer than 200 copies; not enough to warrant revisions for X.3 versions of the DOS.

As far as the XLR8er goes, it works quite well with gate-array Model 4s; even with a graphics board installed. On the other hand, I believe that your two external floppies are 720K drives, not 1.2Meg! The Tandy 2000 used 720K drives (80T, 2-sided). Your floppy disk controller couldn't support 1.2 Meg drives. While you are thinking about the XLR8er, be aware that due to the availability and pricing of memory chips (DRAMs), the XLR8er is no longer available at the currently noted price. We are negotiating a solution to the availability problem with H.I.Tech. What may occur is that the board may become available without memory for a lower price and DRAMs would then be for sale at the current market price, if they are available. I won't know until later on whether or not this method will be workable. I am predicting a price sans RAM of about $225. Current price on 256K-150ns RAMS is about $9-$10 each and the board uses eight of them.

Here's a late breaking update. We worked out a deal; MISOSYS is now the supplier of XLR8er boards. Current price without memory is $175+S&H. As memory prices are still going up , I don't know how long we'll be able to support the $245 price with RAM. Best bet is to call for current pricing. I just was able to acquire 50 chips yesterday (that's April 27th) for $9.80/chip and still had to talk the distributor down from $10.60. Word is that memory prices are still going up. It's a far cry from the $2.75/chip a few months ago for 256K-150ns chips. As of mid-May, I have gotten quotes ranging from $10.60-$15.50 for 50-100 chips.

Does LOF() have a bug?

Fm Scott Ranstrom: PRO-EnhComp's LOF() function always seems to show a different number of records than the directory command of the DOS shows for R-mode files.

Fm MISOSYS, Inc: Scott, concerning EnhComp and the LOF() function, I wrote a small test program which appears to demonstrate to me that EnhComp provides the correct number

of records for a random access file. Here's a copy of the program:

```
ALLOCATE 1
OPEN "r",1,"TESTFILE/DAT:4",20
FIELD 1, 16 AS A$, 2 AS A1$, 2 AS A1$
LSET A$="This is record: "
FOR I = 1 TO 45
A1$ = MKI$(I)
PUT 1,I
NEXT
CLOSE
SYSTEM"dir test:4"
OPEN "r",1,"TESTFILE/DAT:4",20
PRINT LOF(1)
CLOSE
END
```

Of course, the correct number was displayed.

If you have a specific instance where you believe that EnhComp is not providing the correct answer, please provide us with details. A copy of your program on disk is necessary unless it can be demonstrated with a few lines of code. Incidentally, my program was run under LS-DOS 6.3 on a Model 4.

As an afterthought, don't forget that EnhComp will be calculating the number of records based on the LRL at OPEN time. The DOS, on the other hand, will be calculating the records based on the LRL in the directory. Note that a file created by EnhComp with a 78-byte record length will have an LRL of 78 in the directory. That may not be the case with other BASIC's. This point may need clarification.

I just checked Microsoft's interpreter BASIC on a Model 4. It will use the LRL you pass in the OPEN statement for files which it inits, just like EnhComp. MS-BASIC does its own internal blocking and deblocking; it does full sector reads/writes even for records blocked less than 256. EnhComp uses the DOS record I/O facility for records blocked less than 256 LRL. If BASIC creates a file, be it EnhComp or MS-BASIC, where you have used an LRL of 78, for instance, the LRL showing up in the DOS directory will be 78. But either BASIC will permit you to OPEN a file at a given LRL regardless of the LRL in the directory. DOS 6.x will return an LRL fault error if you do indeed open a file with a different LRL than that which is in the directory, but many language programs such as BASIC and MC ignore the error.

The bottom line is to double check that the LRL is the same in both the directory and the BASIC program before suggesting that the LOF() function returns "different" answers.

LB Database

Little Brother-M4 L-50-510

LB is a **flat file data management system** where ease of use is its primary goal; you don't need to program anything or remember complicated command sequences to manage data. Even for the most complex data management needs, LB produces results quickly; EVERY function in LB is menu driven and comes with complete on-line HELP information.

To set up a data base, you just define the record layout. For each field, enter a descriptive name, type, and length. LB handles up to **65534 records**; each can contain up to **1024 characters**. LB supports up to **64 fields per record**; fields may be up to 254 characters long. There are seven types of data fields available: alpha, numeric, right justified, literal, dollar, float, and calculated (add, sub, mul, and div); any of which may be a *Protected Field*, so that its data will not be displayed unless the proper *Password* is entered.

You next establish a *screen*, and you are ready to begin entering data! You may view or edit any record at any time. Find information quickly. You can even create an *index* to your data so any record can be accessed within seconds.

Simply define a print format screen, and LB will print records according to your specifications; 10 different formats can be created. You can print with headers/footers, date, time, page numbering, totals and sub-totals if desired, mailing labels format, and even form letters. You select what records get printed and can use an index for printing in *sorted* order as well; great for organizing your report.

For **automating your processing**, LB can be run in an *automatic* mode; frequently used procedures (such as selecting, sorting and printing) can be saved for future use.

LB requires a minimum of two floppy disk drives and 128K of RAM or hard disk, 64K, and one floppy disk drive).

A Use for LB

Fm Jerry Zimmerman: I just talked to Roy about a question concerning LB and he suggested a note to let people know how I use the product. I bought it with one purpose in mind and that was to make insert cards for audio cassettes with all the song titles listed in the order they appear. I make cassettes and always like to know exactly what's on the cassette. Other people have seen them and now have me make the insert for their cassettes and with LB it is much faster and easier than typing them with a condensed typewriter. This is an excellent data manager. I also use it to find a specific song title on a specific album by title or song length to get the songs for the cassette I am making. I have a VERY large record collection both singles and albums as well as compact discs. I did have to break it into different categories due to the size but now it is manageable and I don't waste hours looking for that song I know should be on this album but isn't.

Thanks Roy and Family for the help and support over the years and I wish you all the best of luck and good fortune in the future.

Feedback from a new LB user

Fm Elmar von Muralt: Dear Roy and Co. At long last I received your parcel with LB, LB-MU PRO-WAM, Lair of the Dragon and The Gobbling Box. Alex' favourite - his best score so far just over 70,000, so it won't be long and he'll be in the vicinity of the record. Please keep us up to date in TMQ what the current record is.

Just a thought for your upcoming elections: Favour the party which is going to bring the U.S. Postal Service into the 20th century; 36 days for an 'AIR PARCEL' is pathetic: maybe they should just get on with moving the mail instead of getting bogged down with umpteen classes of 'Service'. Now that I got that off my chest I feel much better.

Once I had my parcel - after having forked out another eighty-odd dollars for salestax, just another term for highway robbery - I got on to learn about the uses of LB. The manual is quite clear how to go about the various options and I had no difficulty to get results as long as I didn't skip things. Suggestion at the lay out of the manual:

1. Since the various options are not listed in numerical order would it not make sense to arrange a future release in order of building a new data base. e.g.: 10, 1, 8, 9, 2, 3, 5, 4, etc. As I gained confidence with my trial database I got stuck into the 'real' world. Adapting your Basic program listed in TMQ I.i.p68 I converted my /BDA Stock files to /JOB files. Then after defining file and Screen formats the big test - you beauty, it worked like a charm, mind you it took its time, with 1200 recs. in one of the files especially with calculated fields defined. Maybe I should have added the calculated fields later with LB-MU. Just one minor hitch:

2. When using the /JOB file to populate the database, set the screen default number with option 11 else the first record gets messed up. So far I am very pleased with LB and LB-MU and I hope you get many more sales for these valuable packages. I wouldn't be human though not to have a few wishes for enhancements in a future version:

3. I would like to get a second chance when I select a 'run'-option while still having the 'create' disk installed rather than to drop out to LSDOS.

4. Description of 'Select'-option p.99 is not quite clear and could be rewritten to clarify the procedure if one wants to sort by more than one field.

5. Facility to have the same input into the field as in the previous record (handy when most of the info is the same for several records).

6. Display the last record saved in add mode (great when the ruler has slipped away on the original list).

7. Better calculation facilities, maybe a way of calling a user defined subroutine, or maybe a 'look-up' facility.

8. LBMAINT could Have a facility to also upgrade Screen formats in light of the changes made. It would save retyping (a) new screen(s) from scratch.

9. Is there an easy way of duplicating file and screen definitions to other files with identical structure? How 'legal' is it to copy and rename files such as oldname/VDn to newname/VDn. i.e.. is the Filename/LB buried and/or used inside Filename/VDn or Filename/PRn etc.?

10. Facility to merge different databases with identical structure.

Here comes the big one but I suspect I'm banging my head against a brick-wall here:

11. Is there a way to change the date format to dd/mm/yy, preferably in LSDOS as well?

12. Facility to sort by date (maybe we should all change to yy/mm/dd format, at least the computers would have it easier).

The only problem, I am consistently getting, is most likely not even LBs fault: When printing out a report the printer keeps skipping a page: i.e.. instead of moving to the top of the next form it skips one form and forwards onto the top of the next but one form and continues printing there. Do you have any Suggestions?

I can't make any comments about PRO-WAM as yet as I haven't had much done with it so far. Looks like a solution looking for a demand or application. Just one thing popped up, Lynn Sherman's patch for SS is for version 1.00.09. This letter is being written on a registered copy of 1.00.00 and so far it hasn't bombed out. However the point I would like to make, Tandy has not sent me any info about patches and there must be quite a few if there is already at least a version 1.00.09. While I think about it neither have they notified me that TRSDOS6.2 is no longer usable nor have I ever heard anything about patches to Multiplan, both registered packages. They are such a caring lot. Is Tandy maybe run by the US Post Office?

Fm MISOSYS, Inc: Elmar, A month is probably at least two weeks longer than it should take [to get to Australia]. Of course, I don't know where the holdup was. If you can make out any postal cancellation having a date, then compare that to the shipping date in our invoice or the date in the postage meter imprint. Perhaps your Customs held it up.

I will take all your requests to heart. We do expect to work on another release of LB as soon as everything else gets in order here. We are really backlogged in things and have to assess priorities. Our hope is to make some degree of revisions, totally re-write the manual, then offer the revised program and

manual in an upgrade package for a reasonable fee. Yes, the manual is rather disorganized. That's why we are including that little booklet containing TMQ excerpts on LB. If you read that, you will note that it demonstrates the sequences necessary to create a data base operation; it uses a port of mailfile as an illustration. Note that on page 7 (last paragraph), I did state that I set the screen to eliminate the screen prompt using function 11.

I don't know of any patch to change the date structure to YY/MM/DD; I assume you are referring to the ^d^ facility to print the date. I know that outside of the US, you use YYMMDD (or DDMMYY). That's another item for the next release. Actually, when I want to use a date field within a record, I set it up as YY/MM/DD (or eliminate the day and use just YY/MM). In that way, the field is already established for sorting in ascending order by date.

The last question you had concerning LB was about that page skipping. I can't guess why you are getting it. I would suspect that you have either the printer parameters entered wrong or there is an embedded form feed in your print screen. Although you sent a copy of your IC database on diskette, it did not include any print screen file (IC/PR0); if you had sent that, I would most likely be able to answer you. If you send a printout of that screen (the output command) as well as a screen print of the actual print screen), I may have a clue. Or send the file. Also double check any parameters you have set into the FORMS filter. That should be installed. But you may have changed that from 66 lines per page due to your paper length; it's A4, isn't it?

You should be able to access the data file from a language such as BASIC provided the file's LRL is one that BASIC can handle. Yours should be. I am enclosing some information which may be what you need, assuming you can do the programming and interpretation of what is in this information. The definition file may be difficult to read and interpret in BASIC; but once you have the information on the structure of your data file decoded, you may not need to read much of the DEF file.

PRO-WAM is more than a solution looking for a problem, it is a work horse. Mostly everyone has need for a calendar, a note pad, a todo list, etc. PRO-WAM has something for everyone.

As far as Tandy notifying you of patches, most folks from the Australian computer clubs have told me that Tandy has left Australia. They were supposed to take care of the international upgrades for TRSDOS 6.2, according to Bill at LSI.

From your previous correspondence, I don't know of any FAX boards for Model 4 usage; there are quite a handful for the MS-DOS world. That's why I didn't respond to your query.

LB/DEF Files

Fm Daniel L. Srebnick: Are the file layouts available for the /DEF files of LB? It appears that information pertaining to /LB file allocation is contained here. I am attempting to write a program to interface with LB to automate an archive function. I would like to take an LB index file, copy those records alone to an archive file and then delete the records from the first file. The archive file would be added to each time more records are selected for archival and deletion from the first file. This does not appear to be very complicated, as long as the file layouts for the /DEF files are available.

Fm Joe Kyle-DiPietropaolo: Daniel, Shouldn't be too bad, as long as you are working in 'C' or something else that doesn't care too much about the structure of record I/O. It would be tough in BASIC, as the data file structure requires the ability to write one "short" record at the beginning of the file (six bytes as I recall), and then start the beginning of a full record there.

Fm Daniel L. Srebnick: Joe, Can't be at the beginning of the data file, because there is data there. There is a six byte record written at the end of the file. I have a 1000 record file of 256 byte records, that has a trailer record #1001 that ends at byte 6.

Fm MISOSYS, Inc: That contains the number of records in the file.

Fm Daniel L. Srebnick: Joe, I did notice that there is a trailer record of six bytes at the end of the LB files. I have also tracked down what appears to be the number of allocated records and the number of the next unused record in the def file. Actually, there should be no problem doing it in basic either so long as I open the file with the proper LRL for the type of function that I want to perform.

Fm Joe Kyle-DiPietropaolo: Daniel, Those three two byte numbers - the total record count, the first available record and (I think, it's been three years since I looked at the code) the LRL are stored at the beginning of the data file, in addition to in the control file. So, you need to be able to skip the first six bytes, then read in LRL chunks. That means either opening with an LRL of 1 and doing your own blocking, or messing with the open FCB.

Fm Daniel L. Srebnick: Roy, Joe mentioned that at one time LSI had a handout available detailing the file structure of the various LB files. If available, I would like to obtain one. If there is any cost involved, please advise me I have been doing some exploring and see how the number of allocated records is stored, the number of the next available record, and the last deleted record pointer are stored.

But, this is all poking around and not too scientific. Right now I am pondering the structure of the SLx index files. Any

insight on any or all of this you can provide would be appreciated. When I finish my record archiving program I'll forward a copy for inclusion in the quarterly, if interested.

Fm MISOSYS, Inc: That is nothing which was obtained by us. We may get to create such a "package" sometime this year as we will definitely be working on LB revisions this year. All the TRS-80 stuff is out of the way, we have put TMQ II.iii to bed, we're trying to clean our desk up, then get back to the DSM-86 port. After that, it's LB activities. The first job will be to identify those structures - for our own needs.

Fm Daniel L. Srebnick: Roy, Re: my message containing a request for the LB file structure, the necessity for this has lessened, although it would be a nicety at this point. I have dissected the files and think I have determined all I need to know to write my archival program. It is a neat little utility that reads an index file for an input file and appends the indexed records to an output file. Deleted records in the output file are reused, naturally. The output file must already exist and have a matching structure with the input file. Then, the archived records are deleted from the input file. I seem to have been able to maintain full file integrity with my hacking about. I have located pointers for number of allocated records, next unused record number, lrl, last deleted record pointer, and the deleted record count. To your knowledge, are there any other values which I must maintain to ensure data integrity? If you are interested, I would like to submit the program, along with a short article explaining the LB file structure and my program, for publication in TMQ. I understand that you are willing to pay in MISOSYS product, which is fine by me. This little utility helps to automate an otherwise tedious monthly process that I run. I also want to state my satisfaction with LB. It is a superior product for the Model 4 environment.

Fm MISOSYS, Inc: Sure we'll consider the article. I can't respond to the question of integrity without careful analysis of what you've done. After all, I was not the author of LB, and I have not examined it's code to any great extent. I have briefly gone over it to get a rough idea as to how it works. So I'll reserve my response until later.

database.DEF file revealed

Based on a request from Daniel L. Srebnick, of Isle-Net Communications, as to the structure of the LB definition file, we have dug into that a bit and offer you this information. The .DEF file contains the LB structure and the field structure array. The following C program along with the LBDEFS.H header file will illustrate a little bit about the data usable from a program external to LB.

I hope this proves useful to your needs.

```
/* LBDEFS.H
 * This structure defines some of the data from the
.DEF file
*/
struct datares {
     /* The NAME, used to name ALL files */
     char Dbname[16];
     char DUM1;         /* Be sure EOS */
     /* Optional password, encoded */
     char Dbpswd[8];
     char DUM2;
    /* Total record space allocated */
     unsigned short Arec;
     unsigned short Urec;     /* Records in use */
     short Lrl;         /* Total record length */
     unsigned short Ldel;   /* Last deleted record */
     short Nfld;        /* Number of defined fields */
     char Pfld;    /* Set YES if protected fields */
     /* Set YES if unprotected fields */
     char Upfld;
     short Cind;        /* Current index number */
     short Cscr;        /* Current screen */
     /* Screen prior to select request */
     short Cprt;
     short Pwidth;          /* Printer width */
     /* Overlay requesting select facility */
     short Where;
     short Fault;  /* <> 0 if sort/select fault */
     short Ovr;              /* unused */
     short Found;  /* TRUE if database is open */
     /* TRUE if user has password privileges */
     short Level;
     short Auto;   /* TRUE if AUTO mode enabled */
     char Aspec[25];
     long Pos;
     unsigned Adindx;
     unsigned Ttldel;
     int Spare1;
     int Spare2;
     int Spare3;
     int Spare4;
     int Spare5;
     int Spare6;
     int Spare7;
     int Spare8;
};
struct datares LB = {
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0 };
int DATALEN = sizeof(struct datares);
extern int errno;

/* This structure holds info on each individual
field */
struct fldres {
     short Flen;
     char Ftyp;
     char Fprt;
     char Fnam[20];
     };
struct fldres Fn[64];
#define FLDLEN 1536
```

```
/* showdef.c - displays .DEF statistics
*/
#include <stdio.h>
#include "lbdefs.h"
char *prots[2] = {"No","Yes"};
main(argc, argv)
int argc; char *argv[];
{       FILE *fDef;
     if (argc != 2) {
          puts("Syntax: showdef filename.DEF\n");
          exit(1);
     }
     if((fDef=fopen(argv[1],"rb"))==NULL) {
          puts(" Can't open definition file ");
          exit(0);
          }
     errno=0;
     /* read the database definition */
     fread(LB.Dbname,DATALEN,1,fDef);
     /* read the field definitions */
     fread(&Fn[0].Flen,FLDLEN,1,fDef);
     fclose(fDef);
     if(errno) {
          puts(" Error during field load ");
          exit(0);
     }
     printf("\
     Database name                 = %-16s\n\
     Total record space allocated = %-5u\n\
     Total records in use          = %-5u\n\
     Record length                 = %-5u\n\
     Last deleted record           = %-5u\n\
     Number of defined fields      = %-5u\n\
     Contains protected fields     = %-3s\n\
     Contains unprotected fields   = %-3s\n\
     Current screen number         = %-5u\n\
     Number of records in delete chain = %-5u\n",\
     LB.Dbname,LB.Arec,LB.Urec,LB.Lrl,LB.Ldel,\
     LB.Nfld,prots[(int)LB.Pfld],prots[(int)LB.Upfld]
,\
     LB.Cscr,LB.Ttldel);
     exit(0);
     }
```

PaDS

PDS Mod 4 & ATTRIB patch

Fm Gary Phillips: Roy, I considered mentioning this before but decided not to be picky. However, since you've brought up the subject of the PRO-PaDS patch here, it doesn't work. (I was able to fix my copy anyway by just using FED.) The reason is that at one time in the far distant past you provided a patch for one of the modules in PRO-PaDS (back in Notes from Misosys) which required extracting that module, patching it with an X type patch as I recall, and then reinserting it into the library. Anyone who did this now has their library in a different internal sequence from the one you seem to be patching, so the D type patch doesn't work. Probably this hasn't come up simply because there are relatively few PRO-PaDS users, as you've often pointed out. However, I'd suggest that the best way to patch a PDS member (even when the PDS is PRO-PaDS itself) would be by copying the member out to a separate file, patching it there, and then putting it back into the PDS.

Fm Pete Granzeau: Roy, I got a copy of the patch, but it doesn't make sense. I tried examining the file using LSFEDII, and the location being patched does not contain the "F" data at all.

Fm Roy Soltoff: Gary, I hesitate to even look into that. What I may check is whether the patch, as posted in TMQ, patches the master disk as released. That's the only thing I can reference. But yes, the lack of feedback certainly pinpoints the reason for no continued development - not enough sales of the product. When its less than a few HUNDRED, you drop it.

PRO-PADS & date handling

Fm Steven Jerkins: Roy, I seem to remember seeing a patch to Pro-Pads somewhere to fix the date handling. It is no problem but a little annoying to have all the files I have entered into a PDS this year read 1980. Messes up my checking for the latest revision.

On a better side, I really want to thank you for the time and effort you put into the Aerocomp 20meg fiasco with DiskDISK. The last time you heard from me about the problem was just before a lightning strike caused my bubble to crash.

To date, I have received three drivers from Aerocomp; none of which will work with DiskDISK. (version 1.31 was the last). I should be receiving my hard drive back from Aerocomp with a new bubble installed in a few days. They have another driver which is labeled specifically for LS-DOS (with the labeling for LS-DOS I had to pay another 30.00 for it) which the tech on the phone said had all the bugs out. Whichever way it goes I have the patch in hand from the latest TMQ to patch DD/CMD.

Fm Roy Soltoff: Steven, There was never a patch to PRO-PaDS for the "date". The patch you are probably referring to was for the PDS(BUILD) module which exited to an ATTRIB command. That needed tweaking because 6.3 has no USER password (was part of the ATTRIB command string passed to the DOS). The current structure of PaDS allows only a 3-bit year field. It would require lots of leg work to revise. Not enough sales of the product to justify any revisions - as I have said before.

Incidentally, Aerocomp informs me that they have revised their drivers for their hard disks and they now work correctly. They have even tested it out with DiskDISK.

Fm Steven Jerkins: Roy, Does PDS use an offset of 1980 like TRSDOS did? If so I might just Zap the offset date to make a temporary fix. Am I on the right track?

Fm Roy Soltoff: Yes, it uses an offset of 1980.But that would be a tempy solution which would last for only 2 years. After that, the '9' of 1990 wouldn't get output as that is hard coded.

MC - C compiler

MC [DOS 6.x M-21-064] [LDOS 5.x M-20-064]

If you are looking for a **full C compiler**, look no further. If you are looking for a well stocked UNIX System V standard library, look no further. **MC**, reviewed in the January 1987 issue of *80 MICROCOMPUTING*, is a complete C compiler which adheres to the standards established by Kernighan and Ritchie. The library of functions is extensive and System V compatible. The compiler generates Z80 relocatable macro assembler code (M80 or our MRAS). The libraries are files of relocatable object modules. **MC** is a full-featured compiler for the discriminating programmer!

MC supports command line *I/O redirection* for compiled programs, *wild-card* file specifications, parsing for UNIX "." extensions in file specifications, *overlay* support (requires MRAS), a full pre-processor, lots of options, and is designed for the programmer wishing the ultimate in C compilers. The package is supplied with the compiler, pre-processor, an optimizer, assembler macro files, C libraries, a Job Control Language file, the header files, and a 400+ page user manual. MC requires the use of either M-80 or MRAS (available separately), 2 disk drives, and upper/lower case.

dfix() revisited

Fm Pete Betz: Roy: Your Solution to dfix() in the Quarterly was very welcome, as I was still having an awkward time programming around that problem. However, the solution was not without its own complications. It's not possible (with MRAS, anyway) to just assemble the new module and stick it into LIBA/REL, because LIBA contains about 213 modules, and MLIB can't handle more than two hundred! That means it can only be fixed with a patch, and, after much messing around, I think I've got the required specifications:

```
. ZZLIBA/FIX -- By Patch, from source in TMQ 2.3
. for LIBA/REL: corrects failure by floor(),
. dfix() & dint() above 32767.
. command: PATCH LIBA/REL ZZLIBA
.
D1B,14=26 47:F1B,14=24 C7
D1B,3B=61:F1B,3B=49
.
```

This seems to work for me, but take a good look at it before you put it into use, as I'm not all that big an expert....

Fm Roy Soltoff: Pete, We really don't prefer patching REL libraries. We also use MLIB to construct them. However, since that library exceeds MLIB's capabilities, what we do is to prepare the library in two files and then append the second half to the first half using the APPEND command with the (STRIP) parameter. From a user standpoint starting from the big library, you would need to use the splitlib/ccc program which was published in a previous TMQ to split the library into at least two pieces. You then could work on each piece with MLIB and then re-combine the pieces.

Fm Pete Betz: Roy, splitlib is what I used to get at it. But I had no idea sticking it back together was as simple as APPENDing. Thought it was more complicated than that.

Fm Roy Soltoff: Pete, The (STRIP) parameter was added to the APPEND library command of LDOS/LS-DOS specifically to enable you to append two REL libraries.

Fm Mark Harris: Roy, I received my update to PRO-MC today. It looks good and thanks for the quick service. One small nit. The fix to dfix() mentioned in TMQ ii.iii is not included in the update. No problem to me personally. I can do it myself. Still, if you inadvertently left it out, I thought you would want to know.

Finally, I would like to order the SOURCE before you toss 'em out. (Maybe I should just find out what dump your trash is taken to. <grin>)

Thanks a lot for quality products and good support!

Fm MISOSYS, Inc: Your right! I did not get that revision into the master disks yet. Seems like left brain wasn't talking to right brain. I'll get on it and thanks for the reminder. Perhaps its time I try to get someone around here to whom I can delegate that kind of nitty gritty work.

Obtaining "raw" mode

Fm David B. Lamkins: I'm using MC to write an XMODEM program for the Model III. The program opens *CL for update, and needs to read and write raw binary data to the comm port (*CL) using the RS232T driver. It appears that the MC I/O routines are doing some (in this case, unwanted) filtering: A received control-A is translated to EOF, and a transmitted control-I is translated to (I think) multiple spaces. (I base the control-I hypothesis on the observation that the XMODEM "send file" module, which ran properly when compiled under LC, failed when attempting to transmit block #9 [control-I is used as the block number] when recompiled under MC.)

I have searched the manual looking for "magic" options, and have (without success) substituted open() and read() for fopen() and fgetc(). The MC ioctl() function doesn't have a

documented "raw" flag. Is there a simple way to pass raw data through a device using the C library routines, or should I go ahead and write calls to @GET and @PUT?

Fm MISOSYS, Inc: David, Your results are completely predictable. The translation of CTL-A to EOF is the BREAK KEY detection. If you examine the text on the ioctl() function, you will find the reference to the option BREAK control flag. This is device specific. The default is ON which means that a BREAK (i.e. CTL-A on the Model I/III) will be translated to an EOF. You can turn it off via ioctl(). Also, BREAK is covered on page 5-2 under Advanced Topics.

For your second problem, note that CTL-I is a TAB. MC defaults output directed to non-file streams to expand tabs. Again, this is device specific and controlled by the ioctl() function. Page 4-103 actually provides code to change the tabstop. All you need to do to deactivate it is set it to 0.

C vs. BASIC floating point data

Fm Pete Betz: Here's something that I'm curious about. I've been experimenting with making TRSDOS interpreter BASIC and MISOSYS C write compatible disk data files. As one might expect, there's no trouble with integer and string data, but with float and double, they don't automatically match. To make MC write identical data I have to change the first byte of floats and doubles by one.

Well, you may say, the first byte is the exponent and there's a difference in the way the two languages normalize floating points. But the thing that seems strange is that the two variable types disagree in opposite directions. I have to DEcrement the first byte of MC's floats and INcrement the first byte of its doubles to get it to agree with BASIC.

Is this not odd, or am I screwed up, as usual?

Fm MISOSYS, Inc: Pete, that makes no sense whatsoever. MC and M/S use the same number format.

Fm Pete Betz: Roy, Honest! If in BASIC I do stuff like:

```
FIELD 1,4 AS F1$,8 AS F2$
LSET F1$=MKS$(SNGL!)
LSET F2$=MKD$(DBL#)
PUT 1,1
```

then in MC, to get an identical file, I have to:

```
char *t;
t=&sngl;
for(i=0; i<4; i++)
    buf[0+i]=t[i];
--buf[0];
t=&dbl ;
for(i=0; i<8; i++)
    buf[4+i]=t[i];
++buf[4];
write(fn,buf,BSIZ);
```

...until I put those "--" and "++" statements in, the two resulting files couldn't pass a COMParison test. Am I doing something goofy?

Fm MISOSYS, Inc: It surely looks goofy. Guess it's time for me to investigate.

Okay, Pete, I worked up the following small test programs. The first writes out a set of data values in C to a data file. The second one reads those values in BASIC and displays them. When I ran this test, the values displayed were identical.

```
#include <stdio.h>
#include <math.h>
main()
{    FILE *fp;
     double array;
     if (!(fp=fopen("testfp/dat:1","w")))
          exit(-1);
     for (array=1.5; array<11.0; array+=1.0)
          fwrite(&array,8,1,fp);
     close(fp);
     exit(0);
}
```

```
10 OPEN "R",1,"testfp/dat",8
20 FIELD 1, 8 AS FP$
30 FOR I = 1 TO 10
40 GET 1
50 D# = CVD(FP$)
60 PRINT D#
70 NEXT
```

Now then, what's the answer? Of course it dawned on me after I worked all this out and was preparing the *QUARTERLY*. If you look closely at your modifying routine, you will realize that you are altering the least significant bit of the first byte making up the floating point data value. This happens to be the least significant byte of the mantissa. I would certainly expect that floating point algorithmic implementations which are not identical could produce differences in the lest significant bit. However, you will find

that when those "numbers" are converted to ASCII digits, rarely will you find the values to be different. It's possible to find a digit change in the 7th significant digit being caused by that one bit difference; but even that would be circumstantial.

On the other hand, there is a big problem in your adjustment algorithm. All you are doing is incrementing or decrementing one byte with total indifference to overflow or underflow conditions which should be affecting the next most significant byte(s) of the floating point value. For instance, if the original value of the least significant byte were zero, then your decrement would change it to X'FF' which is a considerable alteration. That kind of underflow requires the next most significant byte to be decremented as well. And if it were zero, ...

I would recommend that you forget about modifying the floating point data values. The two algorithms - Microsoft BASIC's and MISOSYS MC's are insignificantly different. After conversion to ASCII, the numbers are equal - within the significance of floating point accuracies. Always remember that it is foolish to compare two floating point numbers for equality. I learned a long time ago in my FORTRAN programming days that the way you compare two floating point numbers for equality was to ensure that the absolute value of their difference was less than some small delta. Look at some engineering programs written in FORTRAN and you will invariably find coding of the form:

```
IF (ABS(REAL1-REAL2).LT.DELTA) ...
```

where DELTA was defined as something on the order of 1E-30 (that's a small value). In fact, if you browse through engineering books on algorithms, you will always see tests evaluating the difference between two numbers being less than delta for ascertaining equality (or whether the two numbers are insignificantly different in value).

Boy, with the discussion on floating point results of EnhComp and this discussion here, it really must be time for me to start working up a series of articles on computer mathematics. How many of you readers want to see that?

Bug in M80/H header file

Fm Lauren D. Harry: I'm having some problems with PRO-MC. I have programs which compile and link without error but produce the following error message when run:

```
** Error code = 16, Returns to X'41F1'
** Illegal logical file number
Open DCB, Device=*KI
Last SVC = 102, Returned to X'1A19'
Std file open error
```

I'm using Microsoft's M80/L80 assembler/linker. The problems appear when my variables are local to main; It seems that the remedy has been to make the variables global. As this is my third week of school, I am very interested in your findings.

Fm MISOSYS, Inc: Lauren, The problem you were having was caused by a missing CSEG in the M80/H header file. Strange as it may seem, MC has been in use for about two years with no one ever reporting that kind of trouble. Certainly only a few folks are using M80/L80 with MC, but one would think that someone else would have uncovered that bug.

It appears that using only globals in main masked the problem by the different sequence of code and data segments which would have been generated. Suffice it to say that you can easily correct the problem by adding a CSEG immediately preceding the *#endasm* statement which is the last statement in the **M80/H** file. Thus, correct the file to read:

```
$HIGH DS    2
       CSEG
#endasm
```

PRO-WAM

PRO-WAM 2.0 M-51-025

This desktop manager gives keystroke access to 4 memory resident **pop-up** applications and disk access of others. A Function Key lets you invoke DOS library commands. PRO-WAM turns your TRS-80 into a powerful machine because it comes with many useful and powerful time savers and desk organizers. Here's some of what you get:

✔ An **ADDRESS** file data base prints cards and mailing labels. Throw away that black book and your Rolodex file.

✔ **HEAD** pipes formatted address data into your letters.

✔ **BRINGUP** tickler file schedules up to 12 items per day by time. New print module. Remember those appointments.

✔ **CAL**endar gives you a month at a glance; covers 4000 years. Flags days with BRINGUP items.

✔ A 3x5 **CARD** filer for a free-form scratch pad of 40 columns by 12 rows. Or use the new **CARDX** with ***forms*** capabilities. It's great for small data base.

✔ **PHRASE** is a KSM from disk for lots of automation.

✔ A telephone list and auto**DIALER** for Hayes modems.

✔ **CALC**ulator gives you 4-functions at your fingertips. **RPNCALC** gives 7-functions in bin, oct, dec, and hex.

PSORT puts your PRO-WAM data files in sort order. *EXPORT* and *IMPORT* functions allow you to move data across windows between applications and programs. There's even an online **HELP** facility!

PRO-WAM works with all programs which use standard DOS keyboard requests and honor the DOS high memory pointer; requires one 32K RAM bank, about 2K of high memory, and a small piece of low RAM. If you have a model 4, then you must have PRO-WAM!

40-CPL video mode

Fm MISOSYS, Inc: Revd. Robert S. Wilson presented the problem of activating PRO-WAM while currently in 40 character per line mode. PRO-WAM doesn't particularly like that and you will see only every other PRO-WAM displayed character. Here's what I did to fix up Revd. Wilson's problem relative to difficulty when activating PROWAM while in 40 cpl mode.

Yes, that would be a problem since PROWAM has always assumed that the Model 4 would be operating in 80 cpl mode. Not to worry, though, because I have developed a patch for PRO-WAM release 2 that will take care of the problem. The patch is WAM22/FIX which follows. What this patch does is preserve the current 40/80 cpl mode state and then force it to 80 cpl mode. This is done the first time PROWAM is activated. Upon exiting PROWAM, the 40/80 cpl mode state is returned to what it was.

```
. WAM22/FIX - Patch to PRO-WAM 2.x Model 4
. only - 03/10/88. Patch allows PROWAM
. activation from 40 CPL mode.
. Apply via, PATCH PROWAM WAM22
D03,8E=7C;F03,8E=64
D03,B3=7C;F03,B3=64
D03,B6=28;F03,B6=10
D0A,DB=72;F0A,DB=53
D0B,71=65;F0B,71=46
D14,35=3A 76 00 32 76 47 E6 FB CD 77 47 18
F14,35=00 00 00 00 00 00 00 00 00 00 00 00
D14,41=D4 CD 53 47 3E 00 32 76 00 D3 EC C9
F14,41=00 00 00 00 00 00 00 00 00 00 00 00
D16,48=69 47 6E 47 73 47
F16,48=00 00 00 00 00 00
. Eop
```

BRINGUP/DAT left open!

Fm Ms. Patricia Scott Garmon: Reference my printout showing the BRINGUP screen and the "Illegal access attempted to protected file" error message. I am getting tired of this error. [I have] used Pronto since you first came out with it. PRO-WAM since October 1987. I Cannot believe I always (almost) exit incorrectly with PRO-WAM when I never did with Pronto. It's inconvenient to constantly have to reset the file and retype my day's entries. Have you a patch?

Fm MISOSYS, Inc: Concerning PRO-WAM and BRINGUP: I can understand your concern when PRO-WAM release 2 *seems* to produce an inordinate amount of cases where the BRINGUP/DAT is left open in contrast to that experienced under release 1. However, this behavior does not necessarily indicate that you have all of a sudden begun to exit *incorrectly*, to use your term. Here's why.

Under release 1, the README file requested you to turn on the NETWORK flag in the DOS to ensure that multiple opening of a single file (as could be experienced if BRINGUP was invoked from BRINGUP) would not cause problems with data integrity. With the network flag engaged, the directory maintains an "open" indicator so that subsequent openings are forced to *read-only* access. If you did not engage the NETWORK flag, secondary access of data files, such as BRINGUP/DAT, would not be forced to *read-only*. But if you did not exit BRINGUP via the BREAK key before powering down, the directory wouldn't indicate that the file was left open. If you engaged the NETWORK flag, it was turned on for all programs - that was a problem to some folks.

Under release 2, we added control of the network flag to PRO-WAM. Thus, the first activation sets the network flag while exit from PRO-WAM restores the flag to whatever state it was in when PRO-WAM was activated. With this operation, the exit procedure identified above as causing no harm under 1.0

(at least as far as the file *left open*) would now show the file left open. At some future point in time, another invocation of BRINGUP would be forced to read-only access.

What I have done to eliminate the problem is come up with a patch to BRINGUP which essentially inhibits any command which will set the "*Changes to file*" facility. This patch inhibits Add, Mov, Echo, Chk, and Del. The patch also will display the letters "RO" at the left of the status line which contains the date whenever BRINGUP is invoked where the data file is in a read-only state. This should solve your problem in that it will keep you from making changes which cannot be filed. Note that you can always BREAK out of BRINGUP if you see the RO indicator and then RESET the file. If you are in the middle of a program, you can always use the LIBEXEC (S-F3) function of PROWAM to issue the RESET BRINGUP/DAT command.

This solution should remedy your problem. For everyone else, the patch is BRINGUP2/FIX which follows.

```
. BRINGUP2/FIX
. Patch to PRO-WAM release 2 - 03/11/88
. Patch inhibits Add, Mov, Echo, Chk, and
. Del commands of BRINGUP if the
. BRINGUP/DAT file is in a read-only
. state. The RO condition is flagged.
. Apply via, PATCH WAM1/APL BRINGUP2/FIX
. Note, the patch can also be applied to
. the provided WAM0/APL library
. provided you have not moved the position
. of BRINGUP in that library.
D26,42=6F 2F;F26,42=D6 28
D28,A6=CD 87 2F;F28,A6=32 B9 2A
D28,CE=7D 2F;F28,CE=90 2C
D28,DF=7D 2F;F28,DF=90 2C
D28,EF=C3 94 2F;F28,EF=7B FE 0C
D2D,6F=FE 29 20 07 11 52 4F ED
F2D,6F=00 00 00 00 00 00 00 00
D2D,77=53 26 2E C3 D6 28 3A 26
F2D,77=00 00 00 00 00 00 00 00
D2D,7F=2E FE 2D CA 90 2C 18 09
F2D,7F=00 00 00 00 00 00 00 00
D2D,87=32 B9 2A 3A 26 2E FE 2D
F2D,87=00 00 00 00 00 00 00 00
D2D,8F=C8 F1 C3 1E 2A CD 8A 2F
F2D,8F=00 00 00 00 00 00 00 00
D2D,97=7B FE 0C C3 F2 2A
F2D,97=00 00 00 00 00 00
. Eop
```

PRO-WAM with the XLR8er

Fm MISOSYS, Inc: L. R. Boatman passed along some difficulties when trying to install PRO-WAM with an XLR8er board installed into his computer along with FIXBANKS, FIXALL, and the RAMDISK modules.

To begin with, when you have all of the XLR8er interface code resident in low memory, that doesn't provide space for the small piece of code which PRO-WAM requires (about 50 bytes). If you use Rex Basham's HIBANKS module in lieu of FIXBANKS provided with your XLR8er, you may regain sufficient low memory to install PRO-WAM. Note that at this time, you must use the FIXALL filter; Rex is revising HIBANKS to add code for handling interrupts which would then eliminate the requirement for FIXALL. [**See the note about the revised HIBANKS in this issue of TMQ.**]

Without the RAMdisk installed, you should not have the problem with PRO-WAM of opening up more than one application. Nothing should prevent that.

If you get strange characters on the screen when you close an application, it is usually the result of defective memory. That's because the screen image is saved into the memory bank which PRO-WAM uses; it is restored to the screen from that bank. Have you been running minimum refresh rate (80 ms) with slow RAM (200ns)? That may be the problem. What bank is PRO-WAM installed into? If its installed into one of the banks 3-10, then you absolutely must have the FIXALL filter installed.

Logical ENTER with PHRASE

Fm Dick Guerin: Roy, I have not had much luck in getting the macro strings working when using the phrase application. The logical ENTER character doesn't appear to function, as of yet - anyway. I've tried both the semicolon and the backslash to no avail. Any help would be appreciated. I would like to remove the KSM/FLT from memory if possible.

Fm MISOSYS, Inc: Give me an example of what did not work for you. The PHRASE application works perfectly for me as well as many others. Also include in your response what program you are trying to export the phrase to and what result you get.

Fm Dick Guerin: Roy, I didn't mean to imply that there was anything wrong with PROWAM. I was wondering what I had done wrong. Anyhow, judging by the reply you made I went back to the manual (glad you were not real short and said RTFM - which I did) found where I had missed the default logical <ENTER> character, CLR-SHIFT-ENTER, and it works (like you said). Thanks again. Why do I always get better answers when you ask the questions?

Fm MISOSYS, Inc: Most folks tend to not ask the question for which they really want to know the answer. You would be surprised as to the kinds of questions we get these days. Sometimes trying to find out exactly what someone wants to know is a job like pulling teeth. Also the kinds of trouble reports leave much to be desired. Things like "it doesn't work" don't really shed too much light on what's happening.

The bank of your choice...

Fm Pete Granzeau: Roy, Is there a way to ensure that Pro-WAM always loads into Bank 2 if it's available, rather than Bank 1? I can force it there with SPOOL, but then when I turn SPOOL off, the spool driver remains memory resident (trapped above PRO-WAM). This brings HIGH$ below X'E800', and I have a frequently used program which loads to that address, with the usual unpredictable results. Reason for wanting: FastTerm will use bank 1 for additional buffer, but not bank 2.

Fm MISOSYS, Inc: If you are using PRO-WAM release 2, then check the manual. Page 10 of the PRO-WAM installation instructions discusses the BANK parameter which is used to direct PRO-WAM to use the bank specified by you. Then a simple

```
PROWAM (BANK=2)
```

will force the use of bank 2 if it is available. If you are only using release 1, then you have to resort to trickery. A previous issue of *THE MISOSYS QUARTERLY* (I.ii) had a program called BANKER which could be used to make-busy a bank or range of banks. Using BANKER, you could make bank 1 busy, load PRO-WAM, then make bank 1 un-busy; again using BANKER. If you have release 1 and have not upgraded to release 2 of PRO-WAM, please rethink your decision. The upgrade offer expires March 31st of this year. PRO-WAM Release 2 is an exceptional product.

Fm Pete Granzeau: Uh, what's the offer? I was unaware there was a second version.

Fm Roy Soltoff: Well, we advertised it in TMQ. We also sent a letter out to every REGISTERED user of PRO-WAM 1.0. Send in your 1.0 master disk with the fee of $24.95 + $5 S&H (US) and you will receive an entire new package. Note that we have let some folks slide past the March 31st cutoff date. But that business ceases on May 31st. **Effective June 1st, the upgrade will cost you $39.95 + S&H**; still a bargain!

Fm Pete Granzeau: I HATE having to send stuff back in! I never have a mailer. Besides, why should I have to register something I bought direct from you in the first place?

Of course, this isn't as bad as the upgrade from LDOS 5.0 to LDOS 5.1, where you had to send back EVERYTHING, including the binder. Cheesh!

Fm MISOSYS, Inc: You have to send your old disk back in because that is what is worth $50 on the trade-in. You don't need a mailer - just an envelope. I don't re-use that disk to send back to you. But a registration card with your name on it doesn't ensure that you are still the current owner. That's our policy, regardless as to where you bought it from.

Don't forget, we also have folks buying products from us to use as gifts to others.

Pro-Wam Developers Toolkit

Fm Bryan Headley: Roy, Please advise as to the status of the Pro-Wam 2.0 Programmers Development Kit. It seems not to come with the generic product (which is nice, and I agree with you on that - but since it did with Pro-Nto 1.0, I assumed it would with 2.0)

Obviously, not much has changed programmatically between 1.0 and 2.0. Maybe you'd like to make an archive with the new Windows/Mac and patches for WinLink? Either that, or a delta with our existing tools?

The library approach is somewhat alien to my way of thinking (what? why merge experimental code into WAM0?). Will get used to it, though... The name WAMLIB has got to go; maybe WinLink... (I ought to see if WamLib is actually PDS is drag!)

Fm MISOSYS, Inc: Release 1.0 tools are sufficient. I published a patch in TMQ for WINLINK. Our feedback was that PROWAM users are not programmers; thus, I didn't want to encumber the manual with stuff for programmers. I also published the "deltas" for the WINDOW/CCC file in TMQ.

You have no need to load APPs under development into a library. That's why PRUN can invoke a stand-alone application - read the manual!

Can't use "Winlink" for the library manager, that is a driver name. And WAMLIB is not PaDS in drag. An APP library is a simplified structure. It has a one-sector directory and fixed length modules. That's needed to simplify the resident code in PROWAM which is used to locate and access a library member.

Fm Bryan Headley: Righteo. Yeah, I remember a patch for WinLink... haven't gone looking for it though. Hard reading a manual when you already know the subject matter, you ever noticed?

A shame. WamLib looks real good; ought to have been the front end to PDS way back when. Not that PaDS would have

sold better (I always thought the concept was partially developed - kept saying to myself, "so?". Perhaps it should have been a PART of something - like EDAS/MRAS.)

Okay, I'll start buzzing through writing that app. Ought to be fun; a REAL cardx app, oriented towards what I do (keep notes on PROGRAMS I'm disassembling)

BTW, I fully agree with your approach to WAMie; PRO-NTO should never have been a programmer's tool (although I would have paid the extra - say, $25, for the toolkit). Sort of diffuses what you are trying to do, saying its the greatest Model 4 program ever (which, even if open to discussion, WAM still comes in the top 3, regardless of the bonehead you are arguing with), and then saying it's as much fun as a Fortran compiler - folks reading the ads get weird when they pick up hints like that...

Fm MISOSYS, Inc: The problem with trying to use a front-end like WAMLIB for PaDS would have precluded direct execution of members without a resident module. The real beauty of PaDS was that /CMD modules were directly invokable. That was the total thrust of its design. Recollect that PaDS was an attempt to satisfy those who wanted to customize their DOS library files. PaDS files were also used in EDAS for *SEARCH; but we could not afford to bundle PaDS with the assembler. We do expect, btw, to get together the toolkit - soon.

The Hardware Corner

Hard drives

ST-506 and R/S driver

Fm Pete Granzeau: I assume the Radio Shack drivers won't drive a Seagate ST-225 (at least, not at 20 meg)?

Fm MISOSYS, Inc: If you are referring to the TRSHD6/DCT or TRSHD5/DCT driver, then they can drive the ST-225 but since it has 612 cylinders and the driver supports at most 404 cylinders, you can only address 2/3rds of the drive. Our RSHARD package with drivers for Model 3 and 4 will support up to 1024 cylinders.

Fm Joe Kyle-DiPietropaolo: ST506 refers to the interface. The ST238 is an RLL rated version of its cousin the ST225, either would be a 20 meg drive when used with the controller from a TRS-80 hard drive. The Miniscribe 3650 is a very interesting drive, with a formatted capacity of about forty megs for under $350 ($399 or less with a IBM XT style controller). The Seagate ST225 should be available for around $225, and about $275 for the ST238 (both without IBM controller).

Percom hard drive

Fm Jack Lottey: Let me say how much I have appreciated the TMQ. I'm strictly an end user working with data-bases and merged word processing to monitor fund raising activities for a nonprofit in the Albany, NY area. I will be moving to the PC/DOS about the first of the year but plan to try to learn something about what makes this Mod III tick, both program wise and hardware wise. I hope I can use some of the excellent software that you have developed to do this. And that I can call on you for a little guidance along the way Thanks, for the info on the driver. Do you know if Percom is still in Dallas on Pagemill Road?

Fm MISOSYS, Inc: Percom is no longer. You may be able to get a question answered from Aerocomp - they bought out the Percom business (I think). Les or Joe here may even be able to respond to a technical question on the 4.

Fm Donald R. Arrowood: Percom is still in business, I picked up a Percom HD and had to get a manual, cable & driver from them and I just called long-distance info and they gave me their phone number - also they go under the name of D. C. Electronic, 3618 Dividend, Garland, TX 75042, Phone: (214) 276-3067. Maybe this is of some help.

Fm Frank Blunda: I found out that D. C. Electronic bought out only the hard drive operation. For anything to do with Percom floppies, the folks to talk to are Computer Serviceland; they are located at 14506-B Lee Road, Chantilly, VA 22021 [703-631-4949]. They supposedly have floppy controller cards.

Hard disk intermittent

Fm Gary Phillips: I have a funny intermittent problem with an early RS 5 Meg drive running LS-DOS 6.3, RSHARD, on a 4P. Once in a while, when I boot the machine, it insists that the HD just isn't there. Normally I boot direct from the HD, and when this happens I get the undocumented message "ID Error" from the boot ROM.

Every time this has happened, if I just turn everything off and wait until the next day then the system works normally again. However, no amount of diddling with cables or connectors seems to solve the problem WHEN it occurs. (I have disassembled the entire unit and cleaned and tightened all connections in one case, but it still didn't work until the next morning!)

On the most recent occasion, I tried something different. Using floppy disk configured to boot the HD, I started with F2 pressed. The DOS booted and then when it loaded CONFIG/SYS it started generating "Error 11H" messages. I could see the HD was being accessed because the activity light was flashing. After letting this cycle for about 60 seconds, suddenly I got the LS-DOS Ready prompt and everything was normal. I then ran Roy's HDCHECK6 on all partitions without finding an error.

When the problem is occurring, I have also tried booting with CLEAR held down, then installing the HD driver interactively. It reports "Drive appears to be unformatted." Still, it works fine the next day. What's going on here? I don't think it's cabling. Should I suspect the 5 Meg bubble itself? The controller? (Not the 4P, I can duplicate the problem on two 4P's and a model 3.) I suspected room temperature, but that has been eliminated. It happens randomly at any temp from 62 to 72 F.

Fm Adam Rubin: Gary, I'm afraid I have no idea what or where the hardware problem is, but I'll offer a guess on what's happening. It sounds to me like the HD/controller can't

find a sector on the directory cylinder, and is returning a "ID not found" error (requested sector ID not found, or ID field CRC error). The 4P boot ROM displays that "ID error" message when this happens, and TRSDOS/LS-DOS would display Error 11H, "Directory Read Error". Being unable to read a sector of the directory could also account for the "Drive appears to be unformatted" message. Again, this is just my guess on the matter, but I hope it's useful.

Fm Gary Phillips: Adam, That all makes sense. HDCHECK6 returns a "Seek error" though.

Floppy Drives

360K floppy disk drive

Fm John G. Gelesh: Recently I purchased a 360K DS DD disk drive for my Model 4. I ordered a Tandon but was sent an IBM drive instead. Does anyone know how to configure this disk drive? I have a Radio Shack cable (with corresponding pins pulled). I want to set this drive up as drive 2. Apparently the IBM drive I have is configured as drive 1 as this is the only way I can select the drive. Can anyone help me? By the way the drive was supposed to be new, but I have my doubts.

Fm Joe Kyle-DiPietropaolo: First off, John, you can't use the Tandy cable for a double sided drive unless you modify it. Is this drive being mounted internally? If so you can do one of three things:

(1) Replace the cable entirely with one that has no pins pulled. (2) Add a new connector at the drive two position with no pins missing. (3) Flip the entire cable over 180 degrees - not end for end, but so that each connector remains in the same place, but is upside down.

For options 1 and 3 above, you will have to modify the Tandy drive you are leaving as the first drive by cutting three of the four drive select traces on the logic board, leaving only the first in place.

IBM drives normally come with the second drive select already installed, sounds like the one you have was changed. You'll have to find the DS trace or jumper and move it. IBM used both Tandon and CDC drives when they were full height drives, so you'll have to do some more describing if you can't find it.

Fm John G. Gelesh: Joe, When I installed my Tandon SS drives in my Model III years ago, they came setup with no configuration (that is, no matter what drive was selected 0, 1, 2, 3 the drive was accessed. I had to either bend out the pins on the programmable shunt or use a Radio Shack cable with the pins pulled. It was very easy to swap drives or change configurations this way. Getting back to my problem with the IBM drive, I must assume that someone cut the traces on the drive for drive select 0, 2, and 3. The drive will only select as drive 1, which does me no good as I already have drives 0 and 1 installed in the computer. Also because of Tandy's curved drive bracket, this IBM drive would be impossible to install as drive 1. I want the drive to be external drive 2 or 3. I do not know whether this drive is a Tandon or not (I am assuming it is). Since I already have drive 1 installed in the computer when I connected this drive to my external cable the controller selected both the IBM and drive 1 drives thereby giving me an Illegal Drive Number error. I have no way of finding out if the IBM drive even works. The cable I am using is a Radio Shack 4-drive (yes, four connectors, that is how I know the drive will select as drive 1, the access light comes on). I can find no evidence of traces being cut. I looked by the card edge connector. Anyplace else to look? How can I get this drive to select as drive 2? or drives 0, 1, 2, and 3 and let the connectors on the cable determine which drive gets selected? There is no terminating resistor pack in this drive as there was on my SS Tandon drives. All there is is an outline where one should go. I don't think it is needed as the only problem I have is selecting the drive.

I tried reversing the drive cable. The computer refuses to boot when I do this. I think everything will work okay if only I can get all drive select lines connected. Any help is greatly appreciated.

Fm Joe Kyle-DiPietropaolo: John, First off, why are you using a four connector cable? If you have a real Tandy floppy controller, you have an external connector for the second two floppies, and the internal connector doesn't carry the drive selects for more than two drives (unless the controller has been modified.

Second, if you were connecting the IBM drive to the external connector, drives three and four are configured for the first and second drive selects, so the drive you have would work as the fourth drive.

Third, there are a set of drive select jumpers on there. Can you find anything labeled DS0 through DS3 or DS1 through DS4? You will find the second set of pins jumpered. You can move this jumper to move the drive select. Of course, this won't help until you resolve the above cable internal/external problem.

Fourth, one of the two possible drives connected to the external drive connector should be terminated.

Fifth, flipping the drive cable over will only work if you also reconfigure you drive selects, as the cable will no longer provide selectivity.

Last, if that's a double sided drive, it isn't going to work with a Tandy cable other than single sided unless you do something about the missing pin 32.

Teac 360K Drive

Fm Bradford A Morse: Can anyone tell me what the strap and/or Jumper setting should be in order to install the TEAC Double-Sided, Half-Height drive as the top drive on a model 4. I'm using the LS-DOS 6.3 with a R/S 5Meg HD. The HD is set-up as drives 0-3 and I'm replacing the original R/S Single-sided drive on top.

At this point, the format command will FORMAT all 80 tracks on a double-sided disk, but then all the tracks are Locked-Out during the 'VERIFY' routine.

Fm Joe Kyle-DiPietropaolo: Bradford, you need to somehow get that cable to pass through the signal on pin 32. Since the pin is missing, there are several alternatives:

(1) Flip the entire cable over, not end-for-end, but so that each connector remains on the same drive, just flipped over. You'll need to modify the original Tandy drive that's still on the bottom so that it only responds to DS0, otherwise it will interfere with the other drive.

(2) Replace the cable with one that has all pins intact. Same as #1 above.

(3) Replace just the connector for the top drive slot. This will not require any changes to the Tandy drive still in the bottom.

In any case, make sure that the new drive is set up for head load with motor on, not head load with select. Which TEAC model is it? A TEAC FD55B is double sided, forty cylinder, while the TEAC FD55F is double sided, eighty cylinder. The FD55G model must be jumpered for "low density only" operation, then its the same as the F.

3.5" Disk Drives

Fm Charles. A. Ainsworth: Roy, There's something I'm not clear about on a couple of items in TMQ on the subject of 3.5" floppy drives. This is more for the record than anything else as I, personally, have no immediate need for clarification.

In the LSI Column of TMQ I.iv, p 44, col.2 ("There have been. . ."), it is stated that due to hardware differences, 720K 3.5" disks on the model 4 would require a completely new driver.

In TMQ II.iii, p 81, under '3.5" floppy drives', Richard Watkins inquires about the usability of 3.5" double-sided drives with 720K disks on the model 4 and will the high density 1.2 Meg floppies also work with it. To which you replied, in essence, that 3.5" floppies will work satisfactorily with the 4, in most if not all cases, as the 3.5" drive emulates a 5.25", but that the higher density disks (1.2 Meg on 5.25" or 1.44 Meg on 3.5") won't.

Your statements seem to be borne out by the letter from Jeff Joseph which follows your answer to Richard. So, is the LSI Column statement incorrect, or was it meant to be limited to the higher density and capacity 3.5" disks?

I am using 80-cylinder double-sided 720K 5.25" disks on my 4's for drives zero through 3, with excellent results, since I long ago dumped the atrociously bad drives that came, first with the III and later with the 4; I don't include the 4D's drives in that statement, they are a little better; and am happy to say that in several years use, I have forgotten what I/O errors are all about. Those are TEAC drives from Aerocomp which I used on the III under LDOS and now on the 4 under LSDOS 6.3. If the 3.5" drives actually emulate those, I don't see how there could possibly be any problem at 720 K.

Fm MISOSYS, Inc: Charles, Let me attempt to clear up the confusion for you and any others mystified by the various statements which have appeared in past Quarterlies.

When 3.5" drives first appeared on the market, some were not totally compatible with their 5.25" counterparts. Since I have no first hand knowledge as to which ones were or weren't, let me not respond to additional questions this statement will generate; I can't comment on brands. What was the source of the incompatibility? That's an easy one. The minifloppy motor is off until it gets a motor on signal. It then starts up and comes up to speed. In a "properly designed" computer, the floppy controller makes use of the ***drive ready*** signal which comes from the drive and which acknowledges that the drive is up to speed. If you attempt to read from a drive which is not rotating at the proper speed, errors will result. Of course the DOS disk driver will be able to circumvent read errors by re-attempts. However, if you try to **write** to a disk before it is up to speed, you risk wiping out data.

The TRS-80 line of computers does not use the "*drive ready*" signal. In fact, that signal line is hard wired to drive select; which means that a floppy driver will see the drive as ready as soon as it is selected. Thus, the floppy disk driver cannot make use of that signal line; no 5.25" drive is instantaneously available. Why doesn't Tandy use that signal line, you ask? Not sure. Perhaps someone more versed in the history of the 5.25" drive may shed some light on that.

The bottom line result is that the DOS (or more precisely, the floppy disk driver) must wait some finite amount of time after enabling the drive motor before I/O is attempted. This waiting time is termed the startup DELAY. LDOS/LS-DOS provide two values for this fixed delay: 0.5 and 1.0 seconds. Early drives needed the 1.0 seconds; drives made perhaps in the last six or seven years start up faster and need only a half a second to get up to speed (wish I only took that amount of time to get up to speed in the morning).

Now back to 3.5" drives. There were some that LSI looked at which would not get up to speed within one second. For those, revisions to the driver would be needed. Also, LSI may have been answering the question in the general sense. Recollect that there were a few different sizes of micro floppies before the world settled on 3.5". Amdek and a few others tryed to get a 3.25" drive accepted (or was that 3.0"?). IBM was ready to go with a 4" drive. LSI did have an Amdek drive on site. That would not work with the regular floppy driver. If LSI was, in fact, including the other micro floppy sizes in their statement, then it was certainly correct. You could not universally state that all micro floppies would work. Perhaps it was better to say that none would. To answer another question, the high-density 3.5" (1.44M) drives were not available at the time of the LSI statement; so don't expect their statement to reflect them.

My statements in TMQ concerning the usability of 3.5" drives were based on present day 3.5" drives which appear to totally emulate a 5.25" 80-track drive. Note that early 3.5" drives were only single sided single density beasts. But today's are different. We have received feedback from folks using various brands of these 3.5" drives with total success using the existing disk driver in LDOS/LS-DOS. I have personally used only one: a Citizen drive which I was having trouble with on my AST 286 machine and I connected that to a Model 4 for test purposes. Sure is far easier to connect one to a Model 4 than a PC.

The high density drives (5.25" 1.2Meg and 3.5" 1.44Meg) can't work on a unmodified Model 4 because the data transfer rate is too high. It may be possible to connect such drives to a machine which uses a switchable floppy disk controller (FDC) that handles 5.25" and 8" drives. That's because I suspect the data transfer rate to be 500KHz, the same as 8" drives; a regular 5.25" drive uses a data transfer rate of 250KHz.

I hope that this lengthy response clears up any confusion which folks may have. For those folks using 3.5" drives in their III's and 4's, if you pass a note to me specifying which brand of drive you used, I will summarize the results for a future TMQ.

Model 4/p Ext. Floppy

Fm Kevin R. Parris: Joe, You mentioned tying pins 6 and 14 to +five volts through 150 ohms. Pardon my low knowledge level, but is this what is called a "pull up resistor"? Can you suggest a convenient spot on the circuit board where I can get the +5 for this? I note in the Model 4 schematic (in the Tech Ref manual) that all of the other FDC interface leads to the drive cable connector are hooked up this way, but every one of them has the same R# reference indicated. Are they all tied to the same resistor? Just for curiosity, what happens if I do not add these connections? The 80Micro article did not mention doing this. I bought two 150 ohm 5% resistors from RS to use for this. I now have two double-sided drives- they are Tandon 100 (as I recall), and one of them has "IBM" engraved on the front, because it was factory installed in a PC. Do these drives mind what position they are in during operation? That is, in the PC they sat flat and the gate opened straight up; would it matter if I had them standing on one side and the gate opened horizontally? Thanks for your help.

Fm Joe Kyle-DiPietropaolo: Kevin, You need each line terminated to a separate resistor, and yes, this is indeed called a pull-up resistor. In the Tandy book they have the same call-out number because they are in a multiple resistor package, not separate discrete resistors.

Both +5 and ground run to diagonal corners of almost every 14 or 16 pin IC in that machine, just pick one nearby. The Tech ref manual will tell you the exact pin for any given IC, there should be a table with this either just before or after the schematics.

Without these pull-up resistors, the leads can "float" into an undefined state when they are not being actively driven. This could result in the drive "thinking" it was selected when actually another drive is selected.

Don't forget to remove all the terminators from the external drives.

The only position floppies don't like is with the logic board down, and that's generally due to convection cooling problems (perhaps long term bearing wear too).

Fm Kevin R. Parris: Joe, I now have two Tandon TM-100 drives working correctly on my 4/P as double-sided 360k devices. I took the connectors off the cable I have, and put pins back in two of them, for the two IBM drives. The other two I left the pins out of, for the two original drives. Everything works just fine. Just exactly what do "terminating resistor packs" look like, and where might I find them on these drives, if they are there, to take them off? Thanks much for all your assistance!

Fm Joe Kyle-DiPietropaolo: A terminating resistor pack on an IBM full-height floppy drive looks like an IC plugged into a socket. Often they are blue, black or brown, and will have markings not typical of an IC, and including the digits 150 or 151 somewhere on there. Usually near the cable connector, often they are the only socketed chip on the board. Just unplug it.

MicroLabs HIRES

Model 4 HIRES graphics

Fm Rob Kraybill: As a new user of a model 4 (I also have a mod I), I am looking to find a high resolution graphics board for my machine. It is a 26-1069 with 128k RAM.

Fm Mark Mueller: Rob, Get the Microlabs Hi-Res board. It's $144 and available from them or Diskcount Data among other places. I don't have the phone # handy but will track it down if you need it. I use the Microlabs board in my 4P with an XLR8er speedup with no problems.

Fm Rob Kraybill: Mark, thanks for the info! I am looking for a cheap (read inexpensive), way to go. Tandy wants $219 for their board, and I can't bring myself to give them that much money for such a little product. Does the Microlabs' board work with Tandy's software? It sounds like what I want, guess I will save my pennies for a while. Thanks again.

Fm Rob Kraybill: Well I finally scraped up the money to buy a Micro-Labs hi-res. board. I am very pleased; it does everything they said it would and more! I am a little confused about the different resolution modes and the difference between the M3 & M4 mode. Do you know of anyone who uses xTCAD? Also do you know of any D.I.Y. A/D Mouse boards other than Micro-Labs'?

Fm Joe Kyle-DiPietropaolo: There is some graphics stuff here [LDOS forum] for the Microlabs board, and some more over on PCS-21, but I'm not that familiar with it myself. As far as the mouse connection goes, somebody was working on a way to connect a Microsoft-style serial mouse, but that wouldn't work with any software that already exists.

Fm Steven Jerkins: I too got the Microlabs board recently. I'm now converting an almost ready for release program to utilize the board. The question is of compatibility with the Radio Shack board. I want the program to check for the presence of a high resolution board and shift modes of display if the board is not present.

Could someone with the Radio Shack board check their documentation and see what ports would be good to access to verify the presence of a radio shack board? Also, anyone have any good ideas as to how to tell if it is a Microlabs or Radio Shack board? (I don't mean opening the case either).

Fm MISOSYS, Inc: Because of the recent increase in questions about graphics boards, and my awareness that more folks have them installed, I asked MicroLabs' Ted Carter to forward me some information on installing their board. So for those having the interest, here's an excerpt from their manual which outlines the procedures necessary to install the MicroLabs board into a Model 4 or 4D (the 4P is a different installation).

Installation and Checkout

Before you grab your screwdriver... As you know, if you haven't opened your Model 4 or 4D computer before, doing so will void your limited 90-day warranty - so you should read through the installation procedure and make sure that you understand what's involved. However, the only real qualifications for the job is the ability to follow directions and use a screwdriver.

Willing to try? Here we go... First, clear a working space about the size of a card table. (It is wise to use a soft, non-static surface to avoid scratching the Model 4 case.) Next, assemble the following tools: 1 Phillips head screwdriver and 1 flat head screwdriver. Unplug the power cord, remove any external cables, and follow these steps:

1. Position the computer on its rear panel to provide easy access to the case bottom and remove the ten screws which hold the case together. (One will probably be covered by a black warranty sticker which you must either remove or poke the screwdriver through it). Note the different types and lengths of screws and their locations.

2. Set the computer upright and remove the black screw from the top center of the back panel of the case.

> WARNING!!! Use EXTREME CAUTION in the next steps when removing and replacing the cover to avoid damage to the Cathode Ray Tube!!! NEVER allow the rear of the CRT to strike any surface, as it may IMPLODE, and cause injury from flying glass!

3. Now, facing the front of the computer, carefully remove the case top by lifting straight up and setting it to the left on its side. Be careful not to pull on the cables connecting the two case halves.

4. Now position the computer so that you are looking at the back panel, with the keyboard furthest from you. You will now be looking at the RF shielding; a flat metal panel with openings in the top. On the latest version Model 4 there is a small connector coming out of the middle of the shield at the top that must be disconnected before the shield will come off. If this is the case, mark or remember the orientation of this connector and then pull up to remove it. Now to remove the panel and expose the main computer board, locate and remove the six screws holding it on. Two screws are located to the right, two on the top, and two on the left side of the panel. The metal shield may now be held in place only by the thick aluminum foil stuck to the bottom of the panel. If it is, then carefully peel this foil off without tearing it and remove the panel.

5. At this point you must determine if you have the latest version Model 4 computer. If you bought your computer prior to September 1984 then you will probably have the original version. In any case, it is easily determined by locating the 34 prong graphics connector. On the original version model #26-1069 it is oriented horizontally in the left, middle of the main computer board and labeled J10. On the new version model #26-1069A and on the model 4D it is positioned vertically in the top, center of the computer board and labeled J12. In the following steps variations in installation required for the "new version" are in parenthesis.

6. Remove the black rectangular jumper between E15 and E14 (refer to Figure 1. located at the end of this section) You may want to save the jumper by pushing it all the way down on just post E14 or taping it inside the TRS-80 case. (On the new version the jumper is located between pins 16 and 18 of the graphics connector J12.)

7. Remove the screw on the main TRS-80 board that is just above and to the left of integrated circuit number U44. (Refer to Figure 1.) There should be a matching screw in the upper, left corner of the Grafyx Solution board with an aluminum spacer crimped onto it. (If you have the new version and there is a screw and spacer in the upper, left corner of the Grafyx Solution board then they should be removed and discarded.)

8. Remove the protective backing from the thick double stick tape located on the underside of the Grafyx Solution board.

9. Push the Grafyx Solution board onto connector J10 (see Figure 1.) (On the new version the board plugs into connector J12 (see Figure 3.) such that most of the board is below the connector.) The connector is very stiff and you will probably have to rock the board back and forth pressing harder at the top and then bottom to carefully work it all the way onto the connector. Be careful not to push too hard on the TRS-80 board. When the Grafyx board is pressed all the way down, about 3/32" of the connector posts will still be visible. Inspect during and after installation to be sure that all 34 of the TRS-80 posts go into the matching 34 holes on the Grafyx board connector. During or after pushing the board onto the connector, you will need to screw in the screw in the upper, left corner of the Grafyx board. This screw gives the board support and helps hold it firmly in place. (On the new version there is no screw or metal spacer to deal with. The Grafyx board is secured by the double stick tape. In addition or in place of the tape, there may be two plastic snap-in stand-off spacers which match up with holes in the TRS-80 board.)

10. Now you are ready to fasten the colored micro-clip to pin 2 of integrated circuit U13 as shown in Figure 2. (On the new version the clip is connected to either of two integrated circuits, both of which are shown in Figure 3. You can connect it to pin 6 of U103 which is the sixth pin from the top on the left side of the integrated circuit. Or to pin 22 of U11 which is the second pin from the bottom on the right side of the integrated circuit.) The micro-clip hook is extended by pressing on the top with your thumb while holding the shoulder in place between your two fingers. Try to place the micro-clip as close to parallel with the TRS-80 board as possible and at the same angle as that shown in the drawing. Be sure that the clip is firmly attached and does not touch an adjacent pin.

11. Now for the moment you've been waiting for! (If you had to remove the small connector at the top of the board in step 4 you will need to re-connect it before proceeding.)

With the cover still off, and making sure that you stay clear of any of the computer electronics, plug in the power cord and turn the computer on. After the video warms up, your Model 4 display should come up just as it did before. Disk owners should have a copy of TRSDOS 1.3, 6.1.2 or 6.2.x in drive 0 and the corresponding supplied Grafyx software disk in drive 1. (Owners with one disk drive or a tape system refer to Section 3A for modified loading instructions.) In response to the TRSDOS Ready prompt, type the following: GBASIC GTEST/BAS. This command will load the Graphics Basic and execute the basic program GTEST. If everything is working properly, the screen will clear, a logo will be drawn, text will be printed in all four borders and different designs will be drawn. If anything looks wrong, immediately turn off the computer! Otherwise, skip over the next number and go to step 13.

Note: If at this point you get the error message "File access denied due to password protection" it is probably because you have removed or changed the normal password on the file BASIC/CMD. To correct this you will need to add the password back or use the BASIC/CMD file from a master TRSDOS system disk.

12. Read this if you have a problem. Every Grafyx Solution board is thoroughly tested before it leaves the factory. Therefore, if any problems in operation are encountered, you should first suspect an installation problem. Removing the board and micro-clip, then installing it again while someone else is carefully double-checking your work will solve most of the problems. Be sure to refer to the drawings and make sure they match up exactly with what you are doing on the TRS-80 computer board. Also make sure the micro-clip is making good contact and only touching the second pin from the top on the left side of part U13. (The sixth pin from the top on the left side of U103 or the second pin from the bottom on the right side of U11 on the new version.) Also be sure that you removed the jumper in step 6.

If you have the new version and the computer does not work at all, make sure that the connector removed in step 4 is plugged back in and in the proper orientation.

If the disk drives don't work properly on the original version computer after putting it back together, take the computer top and shield back off. You will then be able to see a flat white connecting cable located at the top of the TRS-80 board on the

left side. This type of cable connection is very poor and it probably got bumped or bent during installation. First try wiggling this cable in order to get it to make contact. If that doesn't work, remove the cable from the TRS-80 main board and re-insert it. When you do this, be absolutely sure that the conductors line up with the contacts. It is also possible that the cable was pulled loose from the other end, but be warned that if you disconnect the cable from that end that it is very difficult to re-insert.

If you still can't find the problem, write or call Micro-Labs and give a detailed description of the problem and what you have done. And don't worry because we stand behind our product and will get it working for you and if for some very strange reason we can't, we will refund your money.

13. Assuming that everything now works, you can go ahead and finish the job by putting your Model 4 back together. Just to be safe, be sure and unplug your computer and wait at least 30 minutes before proceeding. Since the Grafyx board sits above the main computer board, the metal shield that we took off earlier may not go back in place because it hits the top of the Grafyx board. If this is the case, you can solve the problem very easily. The left side of the shield, which is above the Grafyx board, needs to be bowed outward away from the Grafyx board. This can be done by placing the shield over a table edge and pushing down on the two end corners on the left side. If done properly, there will be 1/4 to 1/2" clearance above the Grafyx board and you will still be able to screw back in all six screws.

14. After, screwing the shield back on, the installation is completed by carefully placing the Model 4 case top over the case bottom. Make sure that all wires are inside the case. AGAIN, BE CAREFUL NOT TO HIT THE CRT NECK SINCE IT COULD IMPLODE OR BREAK OFF. Install the black #6 x 3/8" sheet metal screw in the top, center of the rear panel of the case. Holding the two halves of the case together, carefully turn the model 4 on it's back, as before, and replace the ten screws: five 1" sheet metal screws towards the rear, three 7/8" screws along the front, and two 1" screws in the remaining positions.

15. Turn the computer right-side-up and plug it in! You will probably want to run the test program again just to make sure that nothing got bumped when putting the case back together.

Removing the GRAFYX board

If for some reason you want to remove the Grafyx board, simply reverse the series of steps outlined above. ***The main thing to remember is to put the jumper between E15 and E14*** (between pins 16 and 18 of J12 on the new version) ***back in place or else you will have an inverse video screen.***

Keyboards

Model 4D Keyboard

Fm Ray: Although I tried to order a keyboard replacement for a 26-1069A Mod4 (Gate-Array/clustered keys), National Parts sent me various parts for a 26-1070 Model 4D !

My question - can the "newer" keyboard be used on a 26-1069A machine ? They sent a keyboard,right angle header plug, cable and a resistor pak. The resistor pak has a part number of ARX-0388 and is a ten-pin 820 ohm package - I assume is used for "pull-up" purposes. I'm at a loss as to the location of this resistor pak. My CPU board already has a similar part soldered-in. All other parts seem to "fit."

I'd appreciate any input, especially from someone with a 4D tech manual and/or service manual.

Fm Joe Kyle-DiPietropaolo: Ray, It sounds like you got the complete kit needed to replace the old style mechanical keyboard on an original Model 4. I think that you are all set unless the resistor pack already soldered in your CPU board is of a different value.

Fm Ray: Joe, I called National Parts today and found that the "original" keyboard which looked like a membrane or double-layered conductor-imbedded in plastic is no longer available (it was part AXX-0224) is no longer available. It has been replaced with AXX-0243 which as mentioned is a "mechanical" key arrangement.

Therefore, I have the "correct" parts supposedly, but can't find where the resistor pak is to be used. I suppose it could be a retrofit for the CPU board as you hypothesized, but wonder why National Parts would "market" this part for replacement in a Model 4D (one would already have this installed on the CPU/motherboard.

Once again I appeal to anyone who's either opened up a 4D for Smartwatch or whatever and/or has a tech/service manual for a 4D. The part number again is ARX-0388 and is a 10-pin 820 ohm piece.

Fm Fred Oberding: Ray, The SIP resistor pack (ARX-0388) goes in position RP1, which is adjacent to the connector on the CPU board where the keyboard cable plugs in. The one used with the membrane keyboard is I believe 1500 ohm. I wouldn't bother with it. I changed my keyboard to the newer Model 4D unit and have had no problems. Going the other way,from the mechanical ALPS keyboard to the membrane type would more likely require a resistor change, to prevent erratic keyboard operation. Hope this helps

Fm Ray: Fred, I did leave the RP1 alone (mine was a 820 ohm anyway). After I soldered the cable in position and "fired 'er up", I was back in business.

I guess my problem was that before I soldered anything, I wanted to be sure I had the correct parts (in order to return them if not). Hence, my initial "test" was done holding the cable in place, giving me a not-too-reliable mechanical connection. As stated previously, once soldered in place, I was okay.

Thanks for your help and for joe's comments. I do appreciate the hand!

XLR8er

XLR8er - What is it?

Fm Rob Kraybill: What is the XLR8er?

Fm Joe Kyle-DiPietropaolo: The XLR8er is a speed-up board/extra memory combo for the Model 4. It is available from the manufacturer and MISOSYS. A few people here have them and are quite pleased with them. The only drawback I know of is that TRSDOS 1.3 needs a patch or two to remove some undocumented Z80 opcodes that were used by Tandy.

Fm Rob Kraybill: Joe, what kind of opcodes did Tandy use? RS being what it is I'm not suprised at all to hear this. How fast is the speed-up and how much memory is available? I would imagine that 6Mhz is about the limit without many internal mods.

Fm Joe Kyle-DiPietropaolo: They used some of the cross HL/IX/IY instructions that the Zilog mask does, but the Hitachi 64180 doesn't have them. The clock speed is 6.144 MHz as I recall, and since some instructions have been speeded up there is an additional factor. You get 256K on the board that adds to the existing 128K in the machine for a total of 384K.

XLR8er & 260-1067's

Fm Mark Vandemeulebroucke: Roy, I recently received my XLR8er Board and started installing it today, but immediately entered in some problems. First I find a type number 260-1067 on my Model 4. This type is not mentioned in the installation instructions. There is no RF shield on top of the motherboard. I do not find the 74LS245P chip. There is a 74LS245n chip to the left of the PAL chip (U72) in socket U71. RAM chips are not 150 ns (but this only a temporary problem). Wire W3 runs from U48 to U52, so I suppose it may not be grounded. Will the board work in my computer if I can fix it, connect the cable to the Z80 socket, replace the

RAM chips (2x9 64k RAM chips 150 ns?) I should appreciate Your help.

Fm Joe Kyle-DiPietropaolo: Mark, If you really have a PAL chip in U72 (what numbers are on it?), then that is a non-Gate Array board. Generally, ending letter suffixes aren't that significant on TTL chips (with some exceptions), and a 74LS245N can be considered the same as a 74LS245P. They generally indicate the package type (DIP, PLCC, flatpak and so forth) and will vary from manufacturer.

Is this perchance a foreign language version of the 4? That could account for the stock number difference. I'm guessing that from your name, I thought that I had it rough (check out my full name in the Sysop Roster Bulletin).

Sit down before you price RAM chips, prices have gone nuts in the last six to eight weeks.

Other than that, I can't help you that much with the upgrade, as I don't have one or the documentation for it.

Fm Roy Soltoff: Mark, The different machine number may be due to your international machine. I assume yours is one of the Belgium versions with the international keyboard. On the other hand, I'm surprised that your motherboard does not have a shield. Don't tell me Tandy omitted the shields on the international machines because they didn't have to meet FCC emission standards. In any event, the 74LS245N in U71 is the chip in question. The suffix is irrelevant. On the other hand, I can't comment about wire W3. I'll have to check with HiTech; but they may not have an answer. How old is that Model 4? I am aware of problems in trying to get the XLR8er working in a Rev-A Model 4 (non-gate array). In fact, trying to work it with a graphics board may be harder. We learned of the difficulties with Rev-A boards after your XLR8er was shipped. Now since you referenced that wire W3, you may have the Rev-C or Rev-D board. Why not try the XLR8er first with the machine open. If that's successful, we can try to get you a shield. I'll assume that the "cage" holding the motherboard can accept the standard shield. Joe may know that. I'll get back to you on this.

Fm Joe Kyle-DiPietropaolo: Roy, The sheet metal should be identical, and is available from RS Nat'l Parts. The Fastnut clips will be missing, but that's a local (full service) hardware store item.

What board is best?

Fm Steven Jerkins: Let me ask a few questions if I may. Background on my system. 128K Model 4 (1069A gate array); Microlabs Hi-Res Board; Aerocomp 20 Meg Hard drive (it's Adaptec bubble); Orch-90 board (frequently attached to fascinate the rug-rats); 4 floppies (1 R/S stock, Used PC-XT DSDD drive, 2 Mitsubishi 80trks out of a broken Tandy 2000) (1) Would the XLR8er board or the Alpha Tech megamem board conflict with the equipment I am using now? (not asking for guarantees, just asking for knowledge) 2) Would FDR (Fast Dump & Recovery) as found in the DL's here work on a large ramdisk do you think? 3) Do you of a good way to detect the presence of a HiRes board in a Mod4? I am trying a hand at a large programming task that I want fit for public consumption and would like to detect a hires board and use it if present. (I installed the hires board two days ago and am still learning how to use it)

Fm Joe Kyle-DiPietropaolo: As I recall, the Microlabs board is pretty big, so you might have trouble physically fitting an XLR8er or a Alpha-Tech board in there.

I think that FDR was specifically designed for the 64K ramdisk structures.

Fm Richard VanHouten: As I recall, the XLR8er ads asked you to tell them the exact system configuration, including whether you had a hires board or not. This would seem to indicate that they were able to work around the hires board somehow.

Fm Mark Mueller: I've got a Microlabs board and an XLR8er in my 4P. XLR8er in the modem slot -- uLabs in its connector position. It is REAL tight.

Fm Bob Myers: Steve, I have the Microlabs board, and the Alpha Tech memory in a 4P. They don't have any conflicts. I use FDR6AT (from one of the DL's) with the ramdisk, but I think it is limited to 256K. I use two dumps to load my ramdisk. Don't know about the hard drive, I only run 2 DSDD floppies.

Device timeouts

Fm Gary Phillips: Question about XLR8er and LS-DOS. I sometimes get a device timeout while printing with my XLR8er-equipped 4P. I assume this is because LS-DOS uses a timing loop to decide when the printer has timed out. Of course, the XLR8er shortens such loops considerably. If the spooler is active, the problem will not appear. Can I just patch LS-DOS to increase the device timeout parameter? Or will that make trouble somewhere else. (I have the SOURCE and could probably find the right location.

Fm Joe Kyle-DiPietropaolo: That should work fine. I did the patch up for TRSDOS 6.2 and it is here in DL 6 as PATCH.PTR - the spot to patch has probably changed in 6.3, but you can use this as confirmation of what you've probably already completed.

Fm Roy Soltoff: Gary, I think one of these days I'll put together a patch to tweak the @PAUSE routine. Then things which use @PAUSE for timing (like the printer timeout loop) will function "properly".

Okay, no time like the present. Let me outline a rough plan. Here's the existing @PAUSE code:

```
@PAUSE  PUSH    BC
        LD      A,(SFLAG$)
        BIT     3,A
        CALL    NZ,CDLOOP
        POP     BC
CDLOOP  DEC     BC
        LD      A,B
        OR      C
        JR      NZ,CDLOOP
        RET
```

Since the FAST and SLOW parameters of the SYSTEM command make no sense when an XLR8er is present and running at its fastest speed (for your machine), there is no reason to have the SYSTEM(FAST) test in @PAUSE (that's the SFLAG$ bit test and subsequent doubling of the delay.

There is a couple of ways to deal with the XLR8er speedup. One is to replace the SFLAG$ test with another CALL to CDLOOP. This would be done by replacing the 2nd through 4th lines with:

```
        PUSH    BC
        CALL    CDLOOP
        POP     BC
        CALL    CDLOOP
```

This change would suffice if you have approximately a 50% increase in effective speed of your machine.

Another, more intelligent, way to effect a correction to @PAUSE would require a larger memory space. Since you may have the XLR8er running at different speeds at various times, a "more correct" implementation of @PAUSE is one where the delay loop timing is adjusted at every @PAUSE service call which would correspond to the XLR8er speed currently in effect. Because of memory usage constraints, I think an approximation routine such as the above is workable. It may need some fine tuning for certain installations. For instance, if your machine invariably runs at top speed, you may want to double the delay loop. Here's such a routine:

```
@PAUSE  NOP
        NOP
        CALL    PLOOP
PLOOP   PUSH    BC
        CALL    CDLOOP
        POP     BC
CDLOOP  DEC     BC
        LD      A,B
        OR      C
        JR      NZ,CDLOOP
        RET
```

The reason why an *approximate* rather than an *exact* solution is practical is realized when one recollects my article on SPEED [TMQ II.ii]. Among the various vintages of Model 4s, you will find extreme variation in effective speed. Any Model 4 using TRSDOS 6.2 or LS-DOS 6.3 will be using the same @PAUSE routine even if the machine has an effective speed of 3MHz (the early 4s). So there is no exact science in a DELAY routine which must work across all machines. Approximations are all we can expect. That's a good reason why no programmer should rely on the @PAUSE routine for precise timings.

Another solution which someone may want to look into is to use the 64180's internal programmable reload timers (PRT) for the @PAUSE function. Since those 16-bit timers are impervious to changes in effective machine speed, a timed delay based on PRT values would be constant across different settings of wait states and RAM refresh.

Serious disk problems?

Fm Ford H. McCormack: I recently purchased an XLR8er board from you. After I installed it, I began having serious disk and file access problems. I have just now figured out that this is due to the protection scheme incorporated into LS-DOS 6.3.

I have been a good customer of yours and I expect you to do something about this. Enclosed is my copy of LS-DOS 6.3.

Fm MISOSYS, Inc: Ford, There is no protection scheme imbedded in LS-DOS 6.3 which should cause any problems with your XLR8er board. Had you dropped a line to us before sending off your diskette, I could have saved you the trouble and expense of sending your 6.3 master disk to me as well as save me the expense of returning it to you.

Now let's try to clear up your problems. "*Serious disk and file access problems*" do not shed much light on the actual problem you are experiencing. I do know of two things which may have some impact. First, since the machine is running faster than the DOS expects it to be, certain critical timing loops may need adjustment. We addressed a problem with the floppy disk driver in one of our *THE MISOSYS*

QUARTERLYs. A simple patch to BOOT/SYS will fix the problem of not supporting single density. It's:

```
PATCH BOOT/SYS.LSIDOS (D0D,26=24:F0D,26=12)
```

Another problem has to do with floppy drive delay after select. You may be aware that the DOS supports 0.5 and 1.0 seconds of delay. This creates a delay after selecting a drive before attempting access if the motor was off. Since the delay is controlled by a software timing loop, running the machine at a speed faster than the DOS can allow for can cause it to perform disk I/O before the drive is up to speed. Most folks have used the SYSTEM command to set the DELAY to 1.0 seconds. A simple ***SYSTEM (DELAY=ON)*** will do that for all floppy drives. If your XLR8er is effecting a 100% speedup, that command will then set the DELAY back to a respectable 0.5 seconds.

Still another problem could occur with the sensing of a diskette by the system's CKDRV facility. This function is used to determine if a diskette is in a drive and is turning (wonder where you get the NODISK response from FREE or DEVICE?). Since 6.3 (as well as 6.2) uses a software timing loop, it is possible that the DOS may occasionally not "see" a diskette in a "souped" up machine. I haven't yet investigated that phenomenon since I personally haven't experienced it, although I may look into that soon and report on it. You'll know if your system is prone to that behavior when you start seeing two files with the same name on a disk. When we get around to investigating CKDRV, we'll be reporting on it in TMQ.

I think this should adequately respond to your queries. If you have any other considerations with your XLR8er, please provide greater detail.

Speeding issues

Fm Gary Phillips: I have been using an XLR8er for about a year and a half, so have plenty of space on the RAMdisk for the whole DOS plus utilities plus application files. About six months ago, I added a hard disk and now the speed is even less significant as an issue. Would I trade my 4P for a PC-clone? No way.

XLR8er RAMdrive speedup, et al

Fm Charles A. Ainsworth: Roy, I have installed an XLR8er board in a 4D and am playing around to get the feel of it. I have a couple of questions and a tidbit which may also be of interest to your readers.

But first I want to say how pleased I am with the use of the Ramdisk. Having files up there and loading them is impressively fast, and many are loaded almost before I have time to take my finger off the Enter key. Biggish files like BASIC/CMD, DSM4 and others also load in almost no time. As one of your correspondents once said, I have to dust the carpet lint off my eyeballs when I load a file!

I haven't yet tested HIBANKS (TMQ II.iii, p.37) so I will be talking in terms of the setup as established in the XLR8er manual.

My questions are: First, the XLR8er instructions state that one has to SET *WS FIXALL/FLT and FILTER *KI *WS. In a TMQ you stated that FIXALL/FLT may not be required on some vintages of the model 4, but for now let's assume it is. Now, I like, in many of my applications, to have KSMPLUS active, which means another filtering of the keyboard. With the keyboard filtered twice, once for FIXALL and again for KSMPLUS, will there be any filtering conflict which might produce unexpected or unpleasant consequences?

The second Question: Let's now assume that I only filter the keyboard for FIXALL, without including filtering for KSMPLUS. I then run software that takes over keyboard control, such as the ALLWRITE word processor or many others. Is there any keyboard filtering conflict there?

Now for the tidbit. I'll base my example on the use of the XLR8er for the master LSDOS 6.3 disk I use extensively. After creation of this working diskette with system files, the three Basic files and a number of other utilities, I Sysgen the disk after setting various parameters including FIXALL, SET18O, FIXBANK, etc., then AUTO my DOSTART/JCL file which sets up the Ramdisk banks, backs up my system diskette to the Ramdisk and switches drives so the Ramdisk becomes drive zero.

However, in my first tests of this, I was using the standard format I use for most of my floppies, 80 cylinder, double-sided, with directory on cylinder 40. When the Backup command compared my diskette with the single-sided 71 cylinder Ramdisk with directory on cylinder 1, generated by the creation of ten 32K Ramdisk banks, it of course sensed the differences in format between the diskette and the Ramdisk and went into backup reconstruct, so all the 40 files on my diskette were copied one by one to Ramdisk, S-L-0-W-L-Y and almost painfully, with me fuming at the wait. (There is also another wait while the Ramdisk cylinders are formatted and checked and one sits skewing and watching the 71 dots slowly creeping across the screen and wondering if it is really necessary to check all that RAM every time? I have other utilities for checking RAM but I only use them when I suspect there may be a fault.)

I speeded up the backup of my diskette to Ramdisk by the following: I examined the Ramdisk created with 10 banks and found it has 71 cylinders, single-sided, with directory on cylinder 1, and the name RAMDISK. So I created my diskette with exactly the same features except that I wanted a different disk name to it, SO used XLRDOS. Anyone using other floppy configurations or different number of tracks on their

diskettes or on the Ramdisk must adjust things accordingly. Here's my DOSTART/JCL:

```
. JCL sets up LSDOS master for XLR8 board
SET180 (M=0)
SYSTEM (DRIVE=4,DRIVER="RAMDISK")
E
1
10
Y
ATTRIB :4 (NAME="XLRDOS")
BACKUP :0 :4
SYSTEM (SYSTEM=4)
//EXIT
```

The ATTRIB command sets the name of the newly created Ramdisk the same as the diskette that will be backed up to it, eliminating the prompt for confirmation that BACKUP stops for whenever diskette names are different, and allows a mirror-image backup to proceed without interruption. Incidentally, in case anyone were to think of skipping the ATTRIB command and allowing BACKUP to warn of different disk names and issue the prompt to continue, answering it with Y from the JCL, I found that the JCL aborts at this point, maybe because that's one of the things mentioned under JCL in the DOS manual where "certain BACKUP commands" are excluded from the possibilities of JCL.

Another possibility for getting the Ramdisk set up with one's own diskette name, avoiding the ATTRIB, is to patch the copy of RAMDISK/DCT for each specific application diskette, as follows:

```
PATCH RAMDISK/DCT
(D09,07="XXXXXXXX":F09,07="RAMDISK ")
```

The eight X'es should be replaced with the name one wishes the formatted Ramdisk to have; if the name is shorter than 8 characters, pad it on the right with spaces to make a total of 8. Note that the F section of the patch is RAMDISK plus 1 space.

Users may also have noticed that, also as the DOS manual states, JCL will abort when the change of system drive is made (SYSTEM line of my JCL), so the change of system drive must be the last command in the JCL.

Personally, I prefer to use SWAP/CMD (part of the MISOSYS PRO-GENY package) which allows swapping system drives from a JCL without aborting.

My arrangement allows a mirror-image backup with a remarkable speedup, in as much as complete tracks are being copied uninterruptedly and there is no lost disk-access time looking for individual files. The tracks are copied quickly and the checkup of copied tracks is heart-warmingly fast. I further improved matters by packing the contents of the startup diskette to avoid file scatter and reduce the number of tracks to be copied, by backing up by class from my rough draft disk to my final working disk.

Another thing that gave me some thought: In certain cases I will be needing enough of that Ramdisk to make it advisable to start out with all possible free space. Now, sometimes one overlooks the simple and obvious as I was doing, but after some thought I spotted the fact that quite a chunk of that Ramdisk may be taken up by system files, such as SYS6 (13.5K), SYS7 (7.5K) and/or SYS8 (10.5K). Some analysis, aided by the LIB command in the DOS manual, showed that I didn't really need all those three files up there as I don't use many of those LIB commands in that specific application. Further thought suggested that I could make my own partitioned data set (PaDS) file by extracting modules from SYS files with PRO-CESS and building my own system library file with PRO-PaDS. Voila!, the three files (total 31.5K) got neatly packed into my own SYSLIB file, containing only the LIB commands I need, and occupying no more than some 7.5K.

Then a look at the model 4 Tech. Ref. manual reminded me that I am not using DEBUG from Ramdisk so I omitted SYS5 and SYS9; SYS0 used only for booting; and, of course, SYS13. The total space saving was indeed good.

And now, in lighter vein, in the "not so serious software" department; anyone who is fond of playing GOBBLER as I am when I want to relax, should set up the XLR8er at full speed and then call GOBBLER and see if they can manage to keep up with that!

Fm MISOSYS, Inc: Charles, It's true that I noted in TMQ that FIXALL/FLT may not be required on some vintages of Model 4, i.e. the newer ones. That was based on what the XLR8er manual states - that FIXALL corrects a problem in keyboard scanning. Since I didn't detect any problems on my 4P without FIXALL installed, I made that comment. On the other hand, evidence of late has been uncovered which negates my statement. FIXALL is absolutely required - but not for the reason stated by H.I.Tech. [See next topic for more on this] It seems that the HIBANKS module does nothing to trap the Model 4 interrupts and switch in a lower bank. That's needed! After receiving a few problem reports, I decided to scrutinize Rex Basham's FIXBANKS code. That was derived from HIBANKS. After talking to Rex, we both learned that FIXALL did have code in it to trap interrupts. I have asked Rex to revise HIBANKS to add interrupt trapping code.

In spite of the above, let me address your query concerning consecutive filtering of the keyboard device. I don't see any reason why that should produce conflicts - provided the various filters are correctly coded. The long range problem with too many filters in one device chain is the additive effect

of stack usage. If a program has substantial stack usage and does not switch to an internal stack, there is the possibility of overrunning the system stack. This overflows into the upper end of the device table - usually an inconsequential action.

I am not familiar with ALLWRITE's taking over of keyboard control. I gather it really doesn't *take it over*; on the other hand, it does do something different since it can sense the depression of the CLEAR key. Perhaps it uses the CTL-255 call and notes the keyboard matrix values. Anyway, I'll assume it is not interfering with the keyboard since PRO-WAM is operational from ALLWRITE. Besides, I used ALLWRITE on my 4P with PRO-WAM and an XLR8er board.

Fm Charles A. Ainsworth: Roy, Regarding consecutive filtering of the keyboard, more for clarification than for any action, I would like to state that, when I previously referred to ALLWRITE "taking over" the keyboard, I was actually referring to the KSM-like effect of operating with ALLWRITE which uses Clear as the control key and many alphabetic characters for operational commands plus the numerical keys for user-definable commands. Maybe I used incorrect terminology in talking about "taking over" but then I am not so well versed in the nuances of DOS operation so may have stated it wrong.

Also for your information, the concern I had expressed was due to the following: In my early days with ALLWRITE, I had it on a system diskette which also contained the DOS system and utility files I use for other purposes. This disk was SYSGENed to filter *KI to include my own KSM keys, so when I ran ALLWRITE from this setup I was getting a weird conflict of my KSM keys and the ALLWRITE commands which led me to the conclusion that ALLWRITE should be on a separate disk not SYSGENed to KSM.

Some time ago, when I was first considering an XLR8er, I asked Prosoft whether the use of FIXALL would interfere with the operation of ALLWRITE and they replied that on the model 4, the use of the DOS keyboard driver is mandatory and the use of their ALK driver is not allowed (which I knew and which is stated in their manual) so that any question of FIXALL interfering with the keyboard driver was up to the DOS supplier (I use LSDOS 6.3).

In any event, since my questions first came up, I have lately been running the XLR8er full speed, set up exactly per H.I.Tech's manual, including FIXALL, and ALLWRITE operates perfectly after some rather exhaustive testing, perhaps some 30,000 words in several files. No problem whatsoever. Another thing that concerns me is being sure of reliable disk input and output on a number of large files that must be taken care of. So I have made extensive tests with experimental files under the XLR8er just to see everything goes OK. So far, so good. Just to be on the safe side, I operate with SYSTEM (DELAY) turned on (TMQ, II.iii, p 23, "Running faster...").

FIXALL is needed!

Fm John Coyne: Roy, As a model 4 user I have seen the writing on the wall for some time, however, the way I see it is that my machine does everything I want it to and, with is mid-life update (XLR8 fitted) it does it as fast as I need. I am looking forward to using the model 4 for some time yet. It is therefore that I followed your discussions regarding the way non-profit making software is taken off the market with some interest. I fully appreciate the overheads necessary, even trying to sell the software at reduced price. Now that some of the software has been discontinued would you be in a position to release the commented source of some of these programs, for a price of course. If the source could be distributed on disc and, because they are self documenting your overheads should be minimal. No doubt you will correct me if I am wrong. I do not believe the market for the source code will not be large, however, there may a few other model 4 users who would like to learn a little more and perhaps adapt programs to their own requirement.

Now onto a problem I have been having with SAID version 1.1 (it also occurs with SAID 1.0). I hope you can help. First my configuration: Model 4P, gate array, with XLR8 installed. Fixbank is installed, but I do not use Fixall. I run at (M=1,I=2,R=80) and I have the harddisk driver sitting in high memory.

I have for some time now been running an external clock driver as a background task using task slot 0. The driver has been installed as described in *The Programmers Guide*, therefore I do not believe that the task is the cause of the problem. Furthermore, all programs I have used to date are happy with this configuration. SAID can also work happily with the task installed until I try to use the SAID banking facilities and then I run into problems. I have tried to pin point the problem to the stage where it is repeatable, but without luck. All I have been able to do is get it to fail somewhere in a set sequence of events. When it fails it is normally a system crash. Sequence leading to the problem:

```
Install Ramdisk, set start bank 5, count 5
Using SAID load some source code
Swap the code into bank 3
Load some more source
Swap into bank 4
Load more code
Mark a block and swap with 3
Mark a block and swap with 4
Copy each of the marked block into current
source
Then save all
```

Some times SAID would hang while swapping into the banks, either 3 or 4, other times it force entry to Debug while saving.

The problem does not seem to arise using banks 1 and 2, or banks > 5. To try and isolate the problem I started with a new copy of the master disc and slowly build up my configuration, at each additional module added to the configuration I would stop and test SAID. The problem first appeared after the clock task was added. I have included the source for the clock driver and the full JCL configuration I use. I hope you can help. I can live without the bank switching, but it is a useful facility now that the extra banks are available.

I have two other questions which I hope you are able to answer. Firstly, sometime in the past you gave an indication that you might make some public domain software available using a similar arrangement to Montezuma. Secondly, you also hinted that it may be feasible to update various utilities to cope with the additional instruction set of the HD64180. Could you please provide an update of your thoughts.

Fm MISOSYS, Inc: John, Your problem (program crashes with the bank swapping of SAID when using the XLR8er) is due to your omission of the FIXALL filter. We don't use it either and have had some strange problems just as you report. A few others have reported lockups and crashes under similar circumstances - primarily when an interrupt routine is in high memory.

H.I.Tech erred in not stating that FIXALL was mandatory! Since Rex Basham spent the time to investigate and improve the FIXBANKS utility, I decided to scan over his HIBANKS version and found that it did not handle any banking interfaced to the interrupts. So I called Rex on the telly (that's what you call it, isn't it?) and chatted about that. He looked into FIXALL and found that the filter trapped the interrupts and banked in bank 0 at every interrupt. He didn't know why since H.I.Tech didn't associate the filter with the interrupts - only the Keyboard Scan routine. I clued Rex in as to the necessity of always bringing in bank 0 at each interrupt and he has revised his HIBANKS module to incorporate such a change thereby eliminating the need for FIXALL. In the interim, you will need to install FIXALL/FLT (if you have room) if you are going to use any bank number which is part of the XLR8er board. Look for HIBANKS2 in the next TMQ [this one -ed].

As far as turning loose our source code of discontinued programs, I really feel that it would be a problem. Once I start making my code public domain, folks will get the idea that they can take anything. I have enough problems with theft. Perhaps in my off time I may consider writing another book on Z80 code. I have always considered that. Maybe...

Lastly, the PD issue has been formalized and reported in TMQ II.iii; Vic McClung is chairing that endeavor. Also, that issue of TMQ provided some patches to the DOS to use the MLT instruction. It's a beginning.

Fm John Coyne: Roy, Many thanks for pointing me in the right direction. It should have been very obvious, but I could not see 'the wood for the trees'. The operating system handles the interrupts, but only for the known 128k of memory. It is completely unaware of the XLR8er memory, therefore, it can not switch XLR8er banks to zero if an interrupt occurs whilst addressing one of these banks and we end up with a problem. Anyway, I have added an interrupt trapping routine to my modified version of Fixbank (much the same as Rex Basham's HIBANKS and developed about the same time too). The good news is I can still get all I want in low memory (Ramdisk, PROWAM, and Dduty with a few bytes spare). I have only had a little while to test the system, but it looks good. I will be interested to see what Rex comes up with. The keyboard trap has not been included as I have not had any problem and saw no point in adding more code.

With regarding the source code of discontinued programs, I do not mean make them public domain, but sell the source as you would a program. Copyright would still remain with you. I know this would still not prevent illegal copying, but it is the same situation as it is now. I thought the source may provide you with a little more revenue without having to go through the marketing *ef£ort of a new product. Having purchased *'The Source'* and *'The Programmers Guide'*, I found there was still a lot to learn and the source code of these programs would certainly provide that opportunity. Anyway, I hope your thoughts on writing another book bears fruit. I will certainly be one of your first customers.

Again many thanks for your support. To answer your query on our abbreviations, telly is short for television, and phone is normally used as an abbreviation for telephone. You will be surprised how many American terms are now being commonly used in the UK and even more surprised at just how many differences there are. Reading a considerable amount of literature from the States you become aware of differing expressions and occasionally use them (much to my bosses despair, he tries to keep English English).

Fm MISOSYS, Inc: John, I share your concern that techniques which may be part of our "discontinued" programs may be useful; however, there does exist a burden of overhead in even making the code available. We have (had) many programs. It is a real chore to have to duplicate disks singly. What I mean is that the general work operation of diskette duplication is one of 8-48 copies at a time, depending on sales volume. Our duplication setup has the "masters" residing on a hard drive in the form of diskDISK images. We then dupe to four floppies from the diskDISK image. I really don't have the hard drive space to set aside for images of source code; that would be the capacity of many drives. Most of that would be wasted space. The infrequent requests for source disks would not justify on-line storage. Thus those requests would have to be satisfied by duping from floppy masters. Done one at a time when the need arose would require more manual work. It's not the cost at issue; we would then charge accordingly. But I don't have the time to deal with it. I think we would all be better off if the techniques were made available in book form. That's where my head is. Perhaps one day...

Revised HIBANKS

Fm MISOSYS, Inc: After reading the previous discussions, you realize that we have uncovered some design problems in the HIBANKS module which Rex Basham developed. Let me first take Rex off the hook and state that HIBANKS was designed to be a replacement for FIXBANKS. Thus, it was coded to replace only that module. Now within that design, there turned out to be one flaw. In an attempt to design a faster bank switching routine, Rex chose to use the alternate registers of the Z80. On the surface, this appeared a good choice; it's much faster switching to the alternate registers than pushing/popping a set of regular registers.

Now the DOS doesn't use the alternate registers. It was designed that way in order to free up those registers for application program needs. Some Z80 operating systems reserve the alternate registers for interrupt processing needs since they are faster to switch to; LDOS and LS-DOS do not. On the other hand, it is quite important for anyone programming a module which is DOS-like (a device filter, an interrupt task, an SVC replacement module) to not alter any register unless already documented by the DOS Application Programming Interface (API) as altered. Therefore, HIBANKS needs a revision to avoid using the alternate registers.

Another problem cropped up with folks who did not use the FIXALL filter supplied with the XLR8er software. This was documented by H.I.Tech as being used to slow down the machine during keyboard access. The impression which was left was that it was non-essential. It turns out that FIXALL also switches in BANK-0 (standard high memory) for every machine interrupt. This is essential! The reason is that the DOS interrupt routine forcefully loads bank-0 at every interrupt, and that code won't work if the 64180 has mapped its own memory into the X'8000'-X'FFFF' address region. Anything can happen.

Since Rex had to revise HIBANKS to avoid altering the alternate registers, I asked him to also include code to switch in bank-0 at every interrupt. It makes sense to do it there as most of the code would be already present. Thus, adding a few more bytes of code would then eliminate the need for FIXALL. The revised HIBANKS is included in its entirety on DiskNotes 2.4; this section will include only the changes from the previous version (for illustration of techniques).

Fm Rex Basham: There are block comments on lines 230-237, 329-346, and 368-387 which provide a info on the changes and a corrected description of the memory usage for the upper banks.

New code in the low memory driver begins after the block comment ending on line 387 and extends through line 444. New code in the initialization routine begins with the comments at line 543 and ends with the EI instruction on line 554. The RELTAB addresses and the HBRELx labels are also obviously altered with the exception of HBREL1 & HBREL2. These are still equivalent to the addresses in the original program listing.

I am reasonably sure the new code is bug free but if something else shows up, let me know. I hope this is useful information.

```
;*************************************************
; 5) 03/30/88 - Added code in the driver to handle
;     the system interrupt processor accessed via
;     'RST 38H' instruction.  This was a bug in the
;     original FIXBANKS program which was propagated
;     in the previous version of HIBANKS.
;*************************************************
```

```
;*************************************************
; Low memory routine for handling the @BANK SVC, the
; @VDCTL SVC, and the interrupt handler (RST 38H).
;*************************************************
DRIVER  JR      DVRBGN          ;Standard memory
HIBYTE1 DW      0               ; module header
        DB      3,'$HB'
        DC      4,0
HIBUR   DB      0               ;BUR for high banks
DVRBGN  AND     7FH             ;Strip the high bit
        CP      0BH             ;Banks 0-10 only
        JR      NC,BADBANK      ; else it's an error
        SUB     3               ;Low bank (0-2)?
        JR      NC,BANKGO
;*************************************************
; Switch in bank 0 for low bank requests.  This is
; accomplished by forcing the HD64180 to map Common
; Area 0 out of logical memory, the internal Bank
; area from logical address 0000H to EFFFH, and
; Common Area 1 from F000H to FFFFH.  Thus the 64180
; internal Bank and Common Area 1 are mapped as the
; entire 64K logical address space.  This is the
; exact configuration of the 64180 at start up time.
;*************************************************
        XOR     A
        DB      0EDH,39H,CBR    ;OUT0   (38H),A
        OR      0F0H            ;***** NEW *****
        DB      0EDH,39H,CBAR   ;OUT0   (3AH),A
```

```
;*************************************************
; Common Area 1 is used to map in the logical bank.
; The requested bank begins at logical address 8000H
; and extends for 32K bytes.  The physical address
; is obtained by adding the 4 high-order bits of the
; logical address with a physical offset specified
; in the Common Base Register (CBR). Contrary to
; what I had previously surmised, the memory access
; for the high (3-10) banks does not begin at 128K.
; It starts at 256K of a 512K total memory. The
; memory which would physically reside in the area
; 128K to 256K is not installed. If it were, the
; starting address could be 20H instead of 38H and
; we could gain another 4 logical banks (11-14). If
; someone could determine the hardware configuration
; for the extra 128K, I'd be very interested in
; hearing from them to learn how to do it. I'd also
```

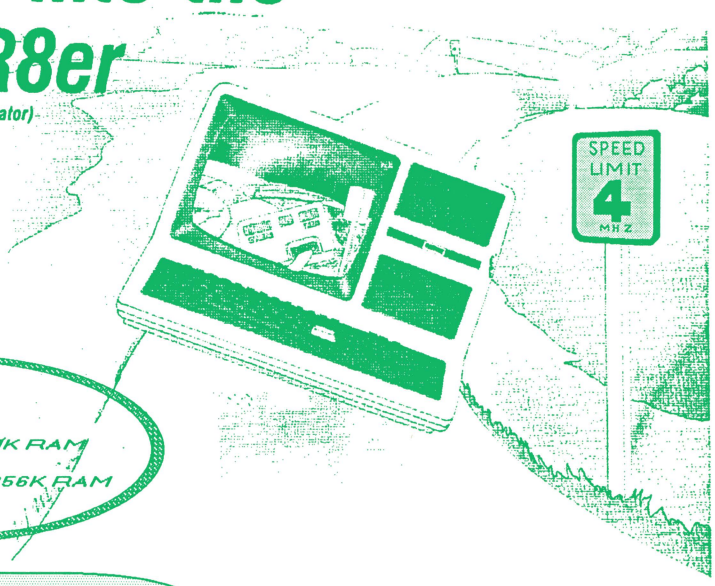
```
; be very happy to work up patches for this program
; and the Ramdisk driver to take full advantage of
; the extra memory.
;************************************************
        ADD     A,38H           ;Bank + 38000H
        LD      B,80H           ;      + 8000H
        DB      0EDH,1,CBAR     ;OUT0   (3AH),B
        DB      0EDH,39H,CBR    ;OUT0   (38H),A
        LD      A,C
        AND     80H
        OR      0               ;Merge in the new
BANKSAV EQU     $-1     ; bank number
        LD      C,A
        BIT     7,C             ;Swap in new bank?
        LD      B,0
        RET     Z               ;No. We're done.
        EX      (SP),HL ;Yes. Transfer to
        CP      A               ; new bank address
        RET
;************************************************
; Up to this point, the code is a duplicate of the
; previous version of HIBANKS/CMD with the exception
; of the INTADDR EQU at the beginning of the program
; and the OR 0F0H instruction flagged with ** NEW **
; in the bank 0 switch above.
;************************************************
; Switch in bank 0 for system tasks.  Since this is
; executed everytime an access is made to the system
; video or the interrupt handler and it's dealing
; with memory swapping, it has to reside in LoMem.
; I opted not to test the current bank to see if 0
; is already resident. The code was 20 bytes longer
; and just a bit slower than automatically doing the
; switch whether or not it's needed. I also replaced
; the EX & EXX instructions with PUSH & POP. It was
; pointed out to me that there are some programs
; which use the alternate registers for something
; other than temporary storage.
;************************************************
; Interrupt switch routine
;
BANKSWI PUSH    BC
        CALL    BANKSW          ;Get MMU config
HBREL3  EQU     $-2
        LD      (IMMUSAV),BC    ;Save it
HBREL4  EQU     $-2
        POP     BC
        CALL    0               ;and off to system
RST38H  EQU     $-2             ; for the interrupt
; Restore previous MMU configuration
        PUSH    BC
        LD      BC,0
IMMUSAV EQU     $-2
        JR      SWEXIT
; @VDCTL switch routine
BANKSWV PUSH    BC
        CALL    BANKSW          ;Get MMU config
HBREL5  EQU     $-2
        LD      (VMMUSAV),BC    ;Save it
HBREL6  EQU     $-2
        POP     BC
        CALL    0               ;Off to the system
VIDEO   EQU     $-2             ; video driver
; Restore previous MMU config
        PUSH    BC
        LD      BC,0
VMMUSAV EQU     $-2
SWEXIT  DB      0EDH,1,CBR      ;OUT0   (38H),B
        DB      0EDH,9,CBAR     ;OUT0   (3AH),C
        POP     BC
        RET                     ;Back to caller
; P/U current MMU configuration
BANKSW  PUSH    AF
        DB      0EDH,0,CBR      ;IN0    C,(38H)
        DB      0EDH,8,CBAR     ;IN0    B,(3AH)
; Replace it with the RESET configuration
; (i.e. Pull in bank 0)
        XOR     A
        DB      0EDH,39H,CBR    ;OUT0   (38H),A
        OR      0F0H
        DB      0EDH,39H,CBAR   ;OUT0   (3AH),A
        POP     AF
        RET
DVREND  EQU     $-DRIVER
```

```
; Fix for the interrupt handler to force bank 0
; resident whenever we're processing interrupts.
; Ditto on the background task note above.
        LD      HL,(INTADDR)    ;P/U the jump and
        LD      DE,BANKSWI      ; driver addresses
HBREL11 EQU     $-2
        DI                      ;To swap the addresses
        LD      (RST38H),HL     ;Jump addr to driver
HBREL12 EQU     $-2
        LD      (INTADDR),DE    ;Driver addr to jump
        EI                      ;We're done
INITRET RET
ICNFG   DW      0
```

TRS 80 MODEL 4 • MODEL 4D • MODEL 4P

Drive your Model 4 into the future with the XLR8er

(accelerator)



The XLR8er provides the following:

- ***Improved speed*** *— up to 8 MHz Z80 equivalent*
- ***Expanded ram*** *— 256KB additional high speed ram memory — optional*
- ***Expanded I/O*** *— optional*
- ***Software utilities*** *— TRSDOS, CP/M, or LDOS — one included with the XLR8er. Additional $15.00*
- ***Simple plug-in installation***
- ***Full one year warranty***

New pricing!
$175.00 with ØK RAM
$245.00 with 256K RAM
(add $5 S&H)

Prices good through August 31, 1988

When ordering, please specify computer model (i.e. 26-1069, 26-1069A, 26-1070) and if HIRES graphics is installed. For 26-1069, note Revision # of motherboard.

MISOSYS Inc.
P.O. Box 239
Sterling, VA 22170-0239

RemBrandt

Complete Business Graphics Toolkit

What did the experts say ?

Computer Shopper Magazine said, *"...you'll be impressed with the drawing and graphic capabilities REMBRANDT puts at your fingertips. It's very powerful, yet easy to learn. Devote a few hours to getting familiar with REMBRANDT and you'll find your good ol' TRS-80 4 or 4P is a handy business graphics tool."*

The **New York Times** said, *"The personal computer makes graphics simple and Spectre Technologies makes a wonderful graphics and drawing program called REMBRANDT."*

What'll it do for me?

REMBRANDT is the only graphic software you'll ever need for your Model 4 computer. Hi-Res board *not* required — it works with the graphics capabilities built into every Model 4, 4P or 4D.

Just look at these advanced features:

Freehand drawing: You can draw lines, boxes, circles, and type on the screen in standard or extra large characters. Full block operations are supported — move blocks of graphic, fill them, copy and delete them and more!

Business graphics: It can read your hand-entered or disk based data and automatically create horizontal and vertical bar charts, pie charts and xy plots with up to three variables. The charts are created on-screen, auto-scaled and labelled — but you can still customize any chart to your specs.

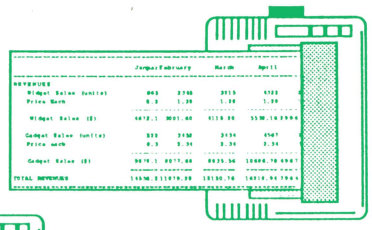
Slide shows: After you've built and saved your graphic screens, put them together for a dazzling on-screen slide show. Move from screen to screen using eleven cinematic special effects like wipes, fades and spirals.

Hard copy: Print your graphic screens on most dot-matrix and daisy wheel printers including the Radio Shack LP and DMP series.

How can I get REMBRANDT?

REMBRANDT is only **$39.95** and is available for the Model 4 (specify the CP/M or TRSDOS/LSDOS version), or get both the CP/M and TRSDOS/LSDOS versions for just $59.95. Add $4 S/H.

LONG & LOUD!



Sideways and Banner Printing Utility for Dot-Matrix Printers

LONG: Did you ever have to print out a spreadsheet that was too wide for your printer? You spent the rest of the afternoon with scissors and tape putting all those little pieces into one useable printout. Well, *LONG* is the answer to your problem... no more cutting and taping. *LONG* twists your printer's output SIDEWAYS and prints spreadsheets (or any text file) of any width the *long* way on one continuous sheet of paper.

LOUD!: Get your message across in no uncertain terms — now you, your computer and your dot-matrix printer can shout ***HAPPY BIRTHDAY MOM*** in eight inch high letters in any of five special type styles. Create banners, signs, posters or oversize greeting cards with ease. Anything you can type can be printed *loudly* in gigantic letters!

LONG & LOUD! works with virtually every dot-matrix printer supporting bit-image graphics including all **Epson** and **IBM** models, **Radio Shack** DMP Models 100, 110, 120, 200, 400, 420, 500, 2100 & CGP-220, **Tandy** DMP-130 and many more.

Available for the Model 4 TRSDOS/LSDOS, all CP/M, and all MSDOS computers. Any single version is just **$34.95**. Any two versions for $49.95! Add $4 S/H.

SPECTRE TECHNOLOGIES INC.
22458 Ventura Blvd., Suite E
Woodland Hills, CA 91364

We accept **VISA, MasterCard, American Express, checks** and **COD** orders. Technical questions, call: 818-716-1655. Order by mail or call our 24 hour toll-free order line: **800-628-2828 (Ext 918)**.

Note: MISOSYS will be closed from August 8th through August 12th

ANNOUNCING

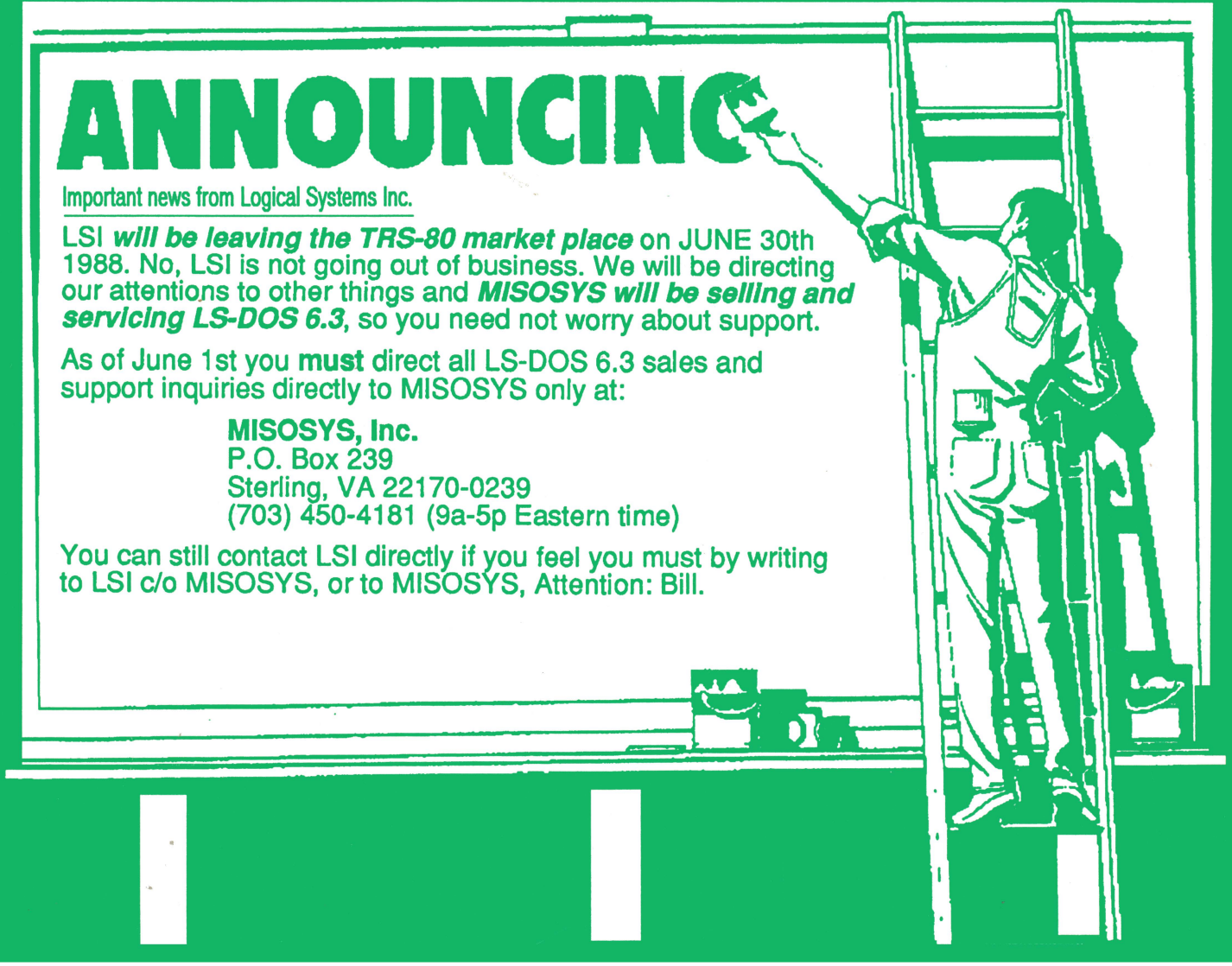
Important news from Logical Systems Inc.

LSI ***will be leaving the TRS-80 market place*** on JUNE 30th 1988. No, LSI is not going out of business. We will be directing our attentions to other things and ***MISOSYS will be selling and servicing LS-DOS 6.3***, so you need not worry about support.

As of June 1st you **must** direct all LS-DOS 6.3 sales and support inquiries directly to MISOSYS only at:

MISOSYS, Inc.
P.O. Box 239
Sterling, VA 22170-0239
(703) 450-4181 (9a-5p Eastern time)

You can still contact LSI directly if you feel you must by writing to LSI c/o MISOSYS, or to MISOSYS, Attention: Bill.



MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170

BULK RATE
U. S. POSTAGE
PAID
Sterling, VA 22170
PERMIT NO. 74