THE MISOSYS QUARTERLY

In this issue:

- Bill Schroeder on licensing
- David Hall on the 64180 CPU
- Gary Phillips on XLR8 and 4P
- Doug Tittle on sorting PRO-WAM data
- Roy Soltoff on WORD with DW II

Volume II, Issue I Summer 1987 $10

THE MISOSYS QUARTERLY

Volume II, Issue i **Summer 1987**

Table of Contents

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

The Blurb

Is it really quarterly?

In case you have already laid in your cord of firewood, let me assure you that we are continuing to put TMQ back on schedule. Here's what happened. The first issue, Summer 1986, was put into the mail on August 19, 1986. Since that was two weeks past the middle of summer, and it takes about two weeks in the postal system, it would have been better titled the Fall 1986 issue.

The Fall 1986 issue was mailed November 28, 1986; that had slipped only one week based on the previous issue's mailing. The biggest problem was the Winter 1987 issue which wasn't mailed until April 14, 1987. That six-week slip in addition to the previous one week slip placed that issue seven weeks behind. The delay was caused by our unusually large workload; we were writing an ED/ASM-86 manual, a DED86 manual, and a RATFOR-86 manual. TMQ had to wait.

The Spring 1987 issue was mailed July 2, 1987, which was two weeks short of a three month interval; thus, we started to correct our schedule. This issue will probably be printed on the DW-II on Sunday, September 13th. It will go to our printer on Monday or Tuesday and we will probably get it back ready for mailing by the 23rd or 24th - missing the end of summer by a few days. But that still pulls us up another two weeks.

We already have a ton of material for the next issue which we now hope to start on in about 4-5 weeks; thus, the Fall 1987 issue should enable us to get back on target. I was almost ready to title this issue the "Summer/Fall 1987" issue, but I relented. I know you will enjoy the material herein; we have every intentions of continuing this publication.

New no-questions-asked RETURN policy

Here's a new policy formally stated in writing which should minimize anyone's fears of trying out one of our products. Effective immediately, any MISOSYS product purchased directly from MISOSYS may be returned within 30 days from the invoice date for a full refund of the product purchase cost (including any applicable sales tax). Returns must be in a "like-new" salable condition and be accompanied by a copy of the original invoice. So if you're not sure whether a MISOSYS product will serve your needs, try it out. If it doesn't fit, return it.

New TRADEIN Policy

Here's another new limited time policy. We will henceforth accept a tradein of any other commercial product as 50% of the purchase of an equivalent MISOSYS product. Got a Model 4 Alcor C compiler and wish you had PRO-MC? Get $62.47 on a trade in. Got an ALDS and wish you had our relocatable macro assembler? Trade it in and get PRO-MRAS for $44.98. Got PFS-FILE or PROFILE and wish you had LB? Trade it in and save $37.47. How about EnhComp at $62.48? Trade in your BASCOM or Alcor MultiBASIC! Got somebody else's ratty disassembler? Trade it in and get PRO-DUCE for half price. Got a Radio Shack Series I (or II) disk assembler. Trade it in for EDAS or PRO-CREATE at $37.48. Want an LDOS 5.3 complete with manual for $32.48 plus S&H? Trade in your DOSPLUS, NEWDOS80, or MultiDOS! Hey, the same thing holds true with our MS-DOS products. We'll take a MASM for 50% off an ED/ASM-86; we'll take a Nutshell for 50% off an LB86; tell us what you have to trade in on a DED86!

New policy on Disk Refreshes

Effective immediately, the charge to recut the disk(s) for any one of our products is $10 +

S&H. The S&H charges are $2 for US, $3 Canada and Mexico, $6 elsewhere. This charge pertains to the same release version. Upgrades are priced separately.

THE MISOSYS QUARTERLY renewals

We have been extremely encouraged by the response rate of our QUARTERLY subscribers' renewal notices. In spite of sending out 500 renewal notices, our current subscription count remains about the same. Although a few folks have dropped out (see Letters to the Editor), new subscribers have made up for that loss. The next batch will be sent out soon.

Incidentally, the renewal statement was implemented as an LB86 print screen from our database. Actually, that task was a little fun; I had hoped to write up the method of preparing such a "form letter" screen for this issue but was unable to get to it. I'll be writing up the screen layout for the next TMQ as it may give a little more insight into LB print screens. The renewal notice was designed as a form letter to print onto a standard 7" statement from NEBS. Since the printer used was a Lineprinter V, the form design had to totally take care of paging. Should make interesting reading for you LB users.

Out of print TMQ's now available

Here's the policy some of you have been waiting for. We are now providing back issues of THE MISOSYS QUARTERLY via copier reprint. The price is $10 plus $2.50 S&H in the U.S. and CANADA. For foreign zone D, the S&H rate is $3.75; zone E is $5.00. That price will be in effect also for regular back issues. What this policy means is that we will no longer be accepting a subscription starting with back issues - only the current issue. Back issues will be available ad infinitum through one means or another. That includes issue I.i and I.ii as well, which are currently out of print.

Projects at MISOSYS, Inc.

We spent too much time implementing PRO-WAM 2.0; but I am sure the results of that effort will be acknowledged by all those who get that release. We are really proud of that one. Of course the work spent doing one thing takes away from the time spent doing another. We did get RATFOR and RATFOR-86 released since last issue. We have also started our port of Disk Sort Merge to MS-DOS for DSM-86. We had completed the first phase - getting a front end to operate on the PC - when we had to drop it to do this issue of TMQ. Within a week, that work will continue.

Articles postponed until later issues

It's great to have more material than we can print. But a lot got deferred from this issue. We had been working on an article discussing the difference in speed of various machines. Starting from a thread on our Compuserve forum, we embellished Adam Rubin's Model 4 speed test so it displayed the calculated run time in seconds. We adapted it to the Model III, as well. We also wrote an 8086 version for MS-DOS. We gathered up five test programs from the July 1987 BYTE which were written in C: Fibonacci, float, sieve, quicksort, and savage. We compiled them for MS-DOS with MSC and for LS-DOS/LDOS with MC. We then ran the suite of programs on two 4's, a 4P with XLR8, a Model III, an IBM PC, an 8Meg PC, and our AST 10Meg 286. But we ran out of space here in this issue. Results in next issue. We should also have our HotShot/286 card installed in our PC by then to give one more set of readings. Just as an FYI, for plain integer work, it appears that the 4P with an XLR8 card was faster than the IBM PC, even a 8Meg Turbo. We also have a page-at-a-time display program from Dan Velting, another UNDATE utility for LDOS from Martin Pollard, a QuickBASIC field input routine from Bill Schroeder, some C programs by Walt Gabriel, and other assorted stuff. The next issue should be coming out sooner than three months. Look for it.

Contents of DISK NOTES 2.1

Each issue of THE MISOSYS QUARTERLY contains program listings, patch listings, and other references to files we have placed on DISK NOTES. Note that to avoid confusion, we have revised the numbering scheme of DISK NOTES to correlate with the corresponding issue of TMQ. Thus, the "2.1" implies TMQ Vol II, Issue i.

Some people enjoy typing in long listings. Sometimes you may have need for only a short patch. If you want to obtain all of the

patches and all of the listings, you may conveniently purchase a copy of DISK NOTES.

There is a cost involved. DISK NOTES is priced at $10 PLUS S&H. The S&H charges are $2 for US, $3 Canada and Mexico, $6 elsewhere. If you purchase the corresponding DISK NOTES with the coupon which accompanies this TMQ issue, you can save $2.50; the cost then being only $7.50 + S&H. Here's what's on Disk NOTES 2.1:

CTERM/CCC	- using C for *CL I/O
FILEFIX/BAS	- Ray Pelzer's BASIC program
FIXES21/TXT	- fixes discussed in TMQ II.i
INITDW2/ASM	- initialize DW-II for PC
INSXBOOT/JCL	- see Gary Phillips' article
MODELB/III	- see Gary Phillips' article
SORTADD/JCL	- see Doug Tittle's article
XBOOT/ASM	- ASM source to MODELB/III

Family Update, by Brenda

Well folks, Roy has asked me to write the family update this time around. You will be able to tell real fast who the writer in this family is. I thought that it would be a nice challenge.

Summer is almost gone. This weekend is Labor Day and as usual we are running a little behind on the TMQ. We still hope to have it out of here by September 21. Next week the girls start their pre-school and I get to start driving the van [Roy's note: Brenda hired on as the school bus driver]. I will still be working in the office in the afternoons. I might even get to sneak in a couple of hours during the morning while the girls are in school. As Roy said in an earlier update, I like to sleep in so it will be a challenge to get up and get going at 6:30 a.m. The last time I did that on a regular basis was when I was in the 7th grade and taking piano lessons. I have a brother and a sister who were also taking lessons, so we had to set up a schedule so that we would all be able to practice our 45 minutes or 1 hour each day. That was well over 15 years ago. I am looking forward to it. Having the office in the house keeps you from getting out and being with other people. I am looking forward to having a little bit more of a social life. The school staff is supposed to have some really good parties.

This summer has been very busy. In June Stacey, Stefanie, & I went down to Miami for a visit with by folks. It was a very busy 10 days. A couple of days we went down to my fathers' print shop and I ran one of the offset presses for a couple of hours. That is the first time I had done that in about 4 years [Roy's note: Brenda used to work in a print shop; in fact, that's where I first met her]. Unfortunately, this was the first time that I had not taken any clothes along that I wouldn't have cared if they had gotten dirty. I was real careful and didn't get them dirty. The girls helped my mother in the office. Stefanie doesn't like loud noises so she didn't like to be around the press.

Just about every evening we went down to the pool. We stayed for about an hour; they never wanted to leave and it would get cool. Once we went down in the morning and the water was actually warmer that the air. Now this is a big pool and not heated.

We got to see a lot of relatives while there. I saw a cousin I hadn't seen in a few years and her little boy. He is about 5. He and the girls got along great. One Sunday morning we went over to the BEACH (I was born and raised in Miami). We went to area that was the "in" spot years ago. It has really changed within even the last five years. Stacey had a great time in the water. Stefanie was content to stay up on the sand (crushed shells) out of the water.

Also, while in Florida we drove up to see my grandparents who live in Lake Placid, FL. We had not seen them since last October when my mother drove them up here for a visit. We were hoping to be able to go out in his new pontoon boat, but it couldn't be delivered until his boathouse was built. There was a problem getting a part for the hoist, so we weren't able to go for a ride. While there my brother and his family from Gainesville, FL. drove down. They have a little boy almost one month younger than Stacey. We all had a great time. There was plenty of room for everyone to sleep. We drove back to Miami on Sunday, since Stacey, Stefanie, & I were due to fly back home on Monday about noon.

Stacey & Stefanie enjoyed the plane rides. They were just short enough so that they didn't get too bored. The flight is just a little over 2 hours. We were one of the first ones off the plane and Roy was right there by the door in the terminal.

July was also a hectic month. We took our boat out just about every weekend. We go to Lake Anna. It is about 80 miles south of us. We got lots of sun and were getting prepared

for our trip to Virginia Beach. This is the first summer that we have had the boat out since we had our little garage fire about 2 1/2 years ago. Stacey was having a great time. She wanted to go fast. We used to take her out in the boat when she was only a few weeks old. We would have her in her car carrier and put her between us towards the bow of the boat. She had a great time. Stefanie is a different story. When Roy was checking the motor to make sure it would start she started screaming "It hurts my ears!" When we got down to the lake and got the boat launched she was okay until we started the motor. She started to scream and then when Roy would speed up she would scream louder. She was hanging on to me for dear life. We would ride for a little while and then stop and swim a little. By the end of the day she was okay. I think part of it was that there were fewer boats so less bouncing around. We went through this just about every time for the first little bit. Of course, this made me wonder how she was going to be at the beach.

During the first week of August my younger neice came for a visit. Last summer her older sister came and spent a few days. This was the other one's turn if we could get her up. She is only 10 and didn't like the idea of flying alone, so it was just a matter of figuring out how to get her up here. When she arrived Stacey and Stefanie were all excited. They had had a great time with their other cousin last year. There are about four girls close to Shanna's age. They seemed to hit it off about from the beginning so she would spend some time with her cousins and then usually after dinner she would do things with the older girls. Since the weather has been hot at times, most of us stay indoors as much as possible during the day. If you looked outside you would think that there wasn't anyone around. However, after dinner everyone, kids and parents, starts appearing outside. There are still a few of us parents that go out and keep a constant eye on the kids especially when they are riding their bikes. It is a great time to socialize.

One evening we had three of the older girls over. Roy made some homemade ice cream and while they waited for that to be ready they played Monopoly. I guess we just got a taste of what it might be like when Stacey and Stefanie get to be that age.

The second week of August we took off for Virginia Beach. We stayed [at the Comfort Inn] right on the ocean. Our room on the 9th floor had a balcony that looked out over the beach. The place we chose to stay was great for us. Since neither Stacey nor Stefanie are much for walking we didn't do any sightseeing. We spent some time every day at the little shops across the way. We must have been located just about in the middle of the strip of stores. Roy & I enjoyed looking around in them. Upon arriving at the hotel we didn't waste any time getting down to the beach. We got there about 4:00 pm. The waves were just right for the girls. At first, Stefanie was very content to just stay up on the beach and play in the sand. Stacey was wanting to go out in the water and jump the waves while holding on to one of us. The weather was beautiful the whole week. We didn't have any rain to speak of and most of the days were no too hot or humid. Each day the waves got a little rougher. In fact, the last two days we were there, the lifeguards were not letting people do much body surfing or use their rafts. By this time, Stefanie was wanting to spend more and more time in the water.

The Labor Day weekend has passed and school has started. After my first two days of driving the bus I am beginning to feel a little more comfortable with it. I am learning my way around areas that I have never been to before. The morning run isn't too bad since I am just picking up the kids. During the noon run I am taking kids home and also picking the afternoon kids up at the same time. I just have to remember who gets off where and when someone gets picked up. Each run takes me just a little over an hour to complete. One of the drivers from last year told me it would be a miracle if I got the noon run done in an hour. That one so far has taken me about 1 1/4 hours to get through. I'm having fun.

Well, I had better stop for now and save a little room for Roy to add something if he wants to.

[From Roy, notice how much MY name got in there?]

Letters to the Editor

THE PROGRAMMER'S GUIDE is available

Fm Merritt Derr: I am looking for a copy of The Programmers Guide to TRSDOS 6.X. Diskcount Data no longer sells the book.

Fm MISOSYS: MISOSYS has copies of the book available. We are the original publisher and hold the Copyright. Our price is $25 + $3 S&H (US). We recently purchased a new Toshiba 5620 copier which is being used for this reproduction as demand is much too small to consider offset printing. As far as we are concerned (and that concern has been in effect for about 1-1/2 years), the market for buyers of THE GUIDE has been too small for us to consider getting another offset print run. Since we now have a high-quality copier, we can provide this last service for anyone wanting a copy.

Fm Shane Dawalt To MISOSYS, Inc:

Ah, a copy of the Programmer's Guide. Now THAT sounds dandy! Are you taking "orders" now?

Fm MISOSYS: Yes, We will duplicate copies of THE PROGRAMMER'S GUIDE TO TRSDOS 6 on an "as needed" basis. We don't expect too many orders. So if you want to order one for the $25 + $3 S&H price quoted, send/call it in.

Here's how to input to THE MISOSYS QUARTERLY

Gary Phillips To MISOSYS, Inc: How do you prefer to receive material for TMQ? In machine readable form, I would imagine... Floppy disk files OK? Program listings and source code as normally used by the compiler/assembler, plus the text of the article in plain ASCII? (CR after each line? or just at the ends of paragraphs? or do you prefer a particular word processor file format?)

Fm MISOSYS: Certainly, TMQ input should be on disk - not paper; I don't have my scanner yet. In the past, we have used ALLWRITE for TMQ preparation. I am beginning to shift over to WORD. This issue of TMQ has been generated using Microsoft WORD on an MS-DOS computer. A plain text file is fine since I would usually have to reformat it anyway to fit into my 2-column format (except for listings). Preferred media is either TRS-80 Model 4 format (one or two sided, 40 track double density) or MS-DOS 360K media format.

Feedback on the last issue

Fm Paul Bradshaw: Got the new TMQ -- congratulations on another fine product!! The 80x86 article was very interesting, and I'd DEFINITELY like to see it continue! I'll be looking forward for my renewal card!

Fm Eben Kent: You continue to do the TRS-80 a great service. Although [not] the most technical individual, I enjoy reading your magnum opus and value your software.

I'm sorry to hear your decision about upgrading LDOS on Model I. Although the decision hurts LNW owners such as myself, I can understand your reasoning. I look forward to the patch to extend dating while I mourn the schism between the Model I and other TRS machines.

I enjoy hearing about your family life. It's one of the reasons my wife lets me subscribe. You do your Philadelphia roots well. As a native, I feel I have a lot in common with you. I'm sorry to say that I did not grow up in the great Northeast, but I spent a good amount of time up there - thanks to Heathkit.

Anyway, continued good luck. I praise MISOSYS publicly whenever I can.

Fm John Cerul: I do appreciate your efforts in providing software for my two Model 4's and the 4P. I realize the market has left me behind, so I do appreciate your support of these machines. In turn, I will endeavor to support your efforts as much as I can. Although much of TMQ is over my head, it is helping me to learn and the more I read it, the more I understand.

I do believe that the penny test (Spring 87, TMQ, p 27) on tires was: If you stood Lincoln on his head, in the tire-tread groove, you were NOT supposed to see the top of his head. If you could see the top of his head, the tread was excessively worn and it was time for new tires.

While you are re-examining Little Brother, I would love to see an easier method of printing multiple copies of one record and print out say 100 copies of the same report or label. I haven't found any data base that allows this easily. The place that this would come in handy is in printing plant pot labels (I'm in the greenhouse business). At present in order to print shipping labels, I route the mailing labels to a file, then pick up that file in Scripsit-Pro, do any adjusting, then print the required number of labels. Rather kludgey at best, no?

Fm Kenneth Peck: I was glad to get TMQ I.iv. I will settle for getting my summer issue in the fall, even a thin issue. Anything to get TMQ on a regular schedule. I started to get worried about you waiting for TMQ. I don't want you to mysteriously disappear one day. Have you considered an option to subscribe to Disk Notes? I don't mean reduced prices or anything. It would mean those who can't afford the time or money perusing CIS could get the LSI and MISOSYS patches a bit sooner.

Since you are considering some upgrading of MRAS, I would like to make one suggestion. How about supporting the "LINES" parameter to the Model 4 version of XREF? I have my printer and FORMS set to print 8 lines per inch, 88 lines per page. It helps keep the clutter around the computer down.

Let me know when the upgrade to MC is ready. I am getting to the point where the patches have patches. Let me also throw my two cents worth in the great MC debate. I make heavy use of both MC and MSC. I find it fairly easy to port code between the Model 4 and MS-DOS using these two compilers.

I never realized that static and global variables were not initialized by MC. Maybe its my background in assembly and COBOL that taught me never to assume anything about uninitiated variables. If I declare a 2K array outside of a function to hold a screen image, I really don't want it using 2K of disk space to store unused zeroes. So I like the way in which MC does not conform. But since that is not K&R, I would suggest a note to that effect in the README file which should prominently state the difference AND how to force initialization of those variables.

If that is the only problem someone has with public domain C code, they are lucky. I have a bunch of that stuff and much of it is buggy, poorly written, and/or written for some very wierd compilers. I usually run it through a C beautifier just to make it legible. Then I have to go through it to do three things. First I have to figure out what their compiler's library is like. Where they have a kinky library function, I have to make it conform to a standard library and/or write kinky functions for my library. Second, I have to remove all the machine and compiler dependent code, of which there is usually a great abundance. Third, I have to clean up the references to header files. Then I run it through LINT with the strictest possible checking to get rid of as many of the potential traps as possible. That is necessary not only when moving PD code to MC, but to MSC as well. Which means, PD software is not always as cheap as it seems.

Regarding <header.h> versus "header.h", you are absolutely right. LDOS/LS-DOS is going to find it no matter what. I suppose you could jazz up MC to start looking for "header.h" on the same drive as the source file before doing the 0-1-... search, but I'm not sure the extra code would be worth the effort. When it comes to moving to another environment, there is a difference. However, my experience is that very little commercial software and PD software observes the difference correctly. Everybody wants me to move my header files where THEY think they belong and to doctor them to contain what THEY think they should contain. Well, I've got news for them... I won't do it. Their code WILL conform to my standard and use my headers or it won't get on my system.

One little bug report on MC. You almost talked about it last time, but it turned out to be something else. I would get an illegal struct/union error on the first line of code

(a comment). How could a comment be an illegal struct/union? Well, simple. I was including a header (z80regs, I think) which made reference to another (stdio, if I recall) which was either not included or included after the "dependent" header. But its not terribly clear in the error message. I probably need to try to recreate the problem.

Fm R. Bains: The long standing bug that was reported in the last TMQ reminded me of a course I attended on Sequential Programming Concepts. The main theme of which was to write CORRECT programs from stage 1 and not to write a program knowingly with bugs in it but unknowingly accepting that bugs will be present and providing patches in the near future to correct bugs. This approach is fraught with one major problem, you can never hope to catch all bugs all the time. To catch 70%, say, means you're doing your job well but 100% bug detection/correction is impossible with this approach. But if you start out and write a correct program you need never correct bugs ever and have perfectly running programs the first time. Is this complete nonsense or is there a grain of sense here?

Fm MISOSYS, Inc: On the issue concerning writing bug free code, I gather that the statements you referenced are sheer nonsense. It is quite easy to say that a bug free program arises when you write CORRECT programs from stage 1; it is quite another to be able to determine what is the meaning of "CORRECT". Bugs materialize in a released program when a given set of conditions established by a user results in a program malfunction, or at least not functioning in the expected and desired manner. It usually is a result of (1) a design error, (2) a design which doesn't take all possible conditions into consideration, (3) a coding error, (4) a coding which produces results differing from the intended design, (5) a macroscopic error where the program conflicts in some way with the operating environment, (6) an error in documentation, (7) omission of a program trap against operator error, or any one of many other kinds of errors. If any organization attempted to produce a PERFECT program, it would be out of business and the program's function made irrelevant years before the program was released. Technology changes much too rapidly for any software program to obtain perfection in its usable lifespan. That's reality. We produce usable programs by making a compromise between the needs of the user for timeliness and the needs of the user for reliability. We programmers do our jobs well when we can fix a problem with a field-installable patch. When we have to resort to a recall, that's when we could have done better.

Fm Tom Wyckoff: A few issues back you printed a short program written in BASIC that showed the inaccuracies of the BASIC interpreter as compared to a compiler. I have been playing with that one lately and getting interesting results. If you run it on a Tandy 1000 it does produce the number shown, about 2700. If basic is invoked by "BASIC /D" it gives the exact correct answer, in about eight minutes. QuickBasic with an 8087 installed does it in six seconds. Converted to Turbo-C, the results were about the same, six seconds. I work for UNISYS, and we sell a large UNIX system called the 5000/90. Next chance I get I plan to run it on one of those. One of our customers that has one is a large aluminum siding manufacturer here in New Jersey. This is no little machine, two 68020 CPUs, two 68001 floating point processors, 8 megs of RAM and a 500 meg disk with 128 terminals around the country. So, what's sitting in an office down the hall? A TRS-80 model 4. They tell me they have several reports they run on it and plan to do so for the foreseeable future. They are still using 6.0.0. I explained what's going to happen to them on December 31, 1987 and gave him a copy of 80 Micro.

Regarding one of the current debates, I enjoy the family news. Although we have never met, I feel as if I have known you for years. As for my "family", Shirley's <the girlfriend> two sons are finally out of high school and gainfully employed. Ginger <the dog> just had her fourteenth birthday. She's slowing down a bit, tail wagging and appetite excluded.

We just finished working in a political campaign. We did the mailing labels for a legislative district. I never thought I'd see the day when both systems, the 1000 and Model 4, were printing labels as fast as we could type them in and I wished we had another machine and operator. That's what I do with all this stuff. Mailing lists and form letters. Next time you take a poll on something, it might be interesting to ask other users what they really do with their systems.

Fm MISOSYS, Inc: Okay, folks, sounds like a good idea to me. Just what do you all do with your systems? We'll publish a representative sample of the feedback in future TMQs.

From Douglas C. Langston: Just a note to give you some of that important feedback that you

are always looking for. THE MISOSYS QUARTERLY is GREAT and the packaging in plastic ensures arrival in a readable condition. Regarding some topics from Volume I, Issue iii, I offer the following:

Page 3, by all means, continue with your family updates, I read and enjoy them and can't help but think how much your family will enjoy reading them in 10 or 15 years from now. What a HOOT that will be!

Page 3, personally I would not be offended by advertisements in TMQ for computer products from sources other than MISOSYS including newsletter availability, particular if all ads clearly stated what operating system that the product is intended for.

Page 4, Disk Notes 7, a page number reference beside the program names would be helpful (I still haven't figured out what HELLO/TXT is about and there is no reference to "Hello world" in the index).

Page 10, PRO-WAM, I would like to suggest that your sales of this product are less than anticipated in part to some confusion with programs such as DoubleDuty and/or Deskmate. At present, I haven't seen a PRO-WAM program or any of the work-a-likes for MS-DOS but will probably buy PRO-WAM after you update it even though I don't know how or when I'll use it. (You can take this as a hint to go into greater detail in a future issue of TMQ as to how PRO-WAM can be used - I've read the review in 80 Micro several times).

Page 24, I also think that TMQ should be advertised somewhere. How about THE CLASSIFIED in 80 Micro at $5/word/issue or $3/word/issue for 3 consecutive issues.

Page 64, LSI's Column is a nice bonus but I am concerned that they are considering their own publication. For cost efficiency, I hope they will reconsider and continue their offering in TMQ even if it means an increased rate.

Quarterly Coupons, how about leaving some space for us to order other items (at regular prices)? It's a great sales tool that isn't being exploited.

Fm MISOSYS, Inc: We still have made no decision about commercial ads; there's not too many other companies around, anyway. The BLURB is the last thing we put together. So far, we really haven't printed out the remaining portions of TMQ by the time the BLURB gets done; thus, the page number references for the notes items aren't available. It's sort of like needing a table of contents. The TOC has been done manually. With WORD, we may consider doing it via the program, but I don't relish merging all of the TMQ .DOC files to do a TOC with WORD. We'll see, the number reference does have merit. Incidentally, the HELLO/TXT file was associated with the "Getting into MS-DOS assembly" on page 84. By now, you may have found the information on the PRO-WAM update which appeared in issue I.iv adequate to embellish that PRO-WAM review. We expect to focus somewhat on PRO-WAM in this and future issues of TMQ. And finally, we have usually added a blank line entry to the coupon; we're looking for all of those orders now.

Fm Larry Rossiter: I enjoy TMQ very much; it provides so much excellent information that cannot be obtained elsewhere. Also enjoy reading about your family, if others don't they can skip that part. My 4 'kids' range in age from 35 to 45 so your stories take me back a few years!

In addition to my Model 4 I have (and use) two Model I computers, and an LNW, so will be in the market for some products as long as you produce them. However, I fully agree with your position on supporting this rather ancient beast; I wonder if some of your readers expect Model I support into the next century.

I purchased LB from LSI some time ago, and MU (bless it!) from you. After considerable use, I find there is one change I would really welcome; on completion of printing out a file, being given the choice of returning to the main menu (as it does now), or going to the 'Enter print format number' prompt in order to print out another file. As I usually print several files in succession this would be quite a time-saver. I am not impressed with the manual but imagine a rewrite would be expensive and would not result in increased sales as one doesn't get to see the manual until after they have purchased the product!

Fm MISOSYS, Inc: Although revisions to the LB and LB86 packages are desired, we don't foresee the time materializing for that project until next year. But we are committed to improving that package with both add-ons like the LBMU package and revisions to LB itself. It will just take time. Bear with us.

On the @EXMEM utility

Fm Steve Woicik: Thank you for the @EXMEM utility in the Winter issue of TMQ. This is the single most useful program for the 128K Model 4. Until now, only the most skilled programmers could access the extra 64K for their own use. I am sure that if @EXMEM had been available when the Model 4 first hit the market, many more Model 4's would be in use today. After all, it is disappointing to discover that the only user use for the extra memory is as a 64K MEMDISK. [editor note: surely PRO-WAM is as useful, if not more so, than a 64K memdisk].

My use for @EXMEM would be to move larger blocks of data in and out of protected BASIC memory. Using the 256 byte page option is like using a random access disk file. A 256 byte subsection of data is transferred with each command. A better use might be for @EXMEM to move user sized blocks of data. Blocks could be adjusted from 1 byte to 16K bytes. This would greatly add to the flexibility of @EXMEM.

It looks to me like @EXMEM could be modified to move any size block of data. I noticed that program lines 160, 192, and 203 load the page length of 256 for the LDIR instruction. I experimented with page lengths of 80, 500, and 1012 bytes by patching the lines directly. It apparently works. However, larger page lengths cause the program to crash. Also patching the page lengths into the program for each page size is inconvenient.

Could you modify @EXMEM to move user adjustable sized blocks of data? I am sure there are a large number of Model 4 users who could also benefit from such a modification to your already fine program.

I also have a few questions about @EXMEM that I wonder if you could answer for me: (1) Can @EXMEM be made a permanent part of the operating system so that @EXMEM will load when the DOS is booted? (2) How many extra RAM banks can @EXMEM access? Which of the 3 memory add-on boards would you recommend for use with @EXMEM? (3) Is there a method to directly and rapidly load data from disk to the extra memory and from memory to disk?

Fm MISOSYS, Inc: The @EXMEM interface automatically does double buffering of page transfer; that is necessary in case your buffer is above 8000H. The extra buffer which @EXMEM uses can handle a maximum of 256 bytes since it is located at 2300H. Any size larger than that would crunch any program which was executing at an origin of 2400H (the true lower limit for the library overlay region). You will also note by direct examination of the register protocol that a page transfer uses all of the primary registers. We don't like to use index registers for parameter passing and the secondary (primed) registers are off limits. It takes up more code space to use a register as a pointer to a data parameter area. The design of service functions is always a compromise between maximizing capabilities and minimizing memory utilization. I felt that a fixed page size of 256 bytes was something everyone could live with. Any user-defined memory transfer could be implemented in the calling program easily without taking up valuable low-memory space. @EXMEM could become a permanent part of the system configuration by a SYSGEN. That would only require that the config table of the SYSGEN command be altered to include saving another piece of the SVC table. Now since @EXMEM uses the @BANK service call, therein is the dependency. Any memory add-on which enhances @BANK to support the extended memory add-on (like our patches to the Alpha Tech board or the software included with the H.I.Tech XLR8 board) would be addressable via @EXMEM. Any add-on memory board which provides only a ramdisk driver would not. There are methods to rapidly transfer file sectors to/from memory pages. Check out FDR which is available on our Compuserve forum. I believe it can fast load/save the normal extended 64K. It should be adaptable to other uses.

Corrections, corrections, corrections

Fm H. Brothers: I just finished reading through TMQ I.iv -- as usual, a masterful job! However, I think maybe you confused me and John Harrell. He wrote the review of PRO-NTO in the Nov 85 issue of 80 Micro. I reviewed PRO-NTO about the same time for Online Today. No problem about the mis-reference, though, as far as I'm concerned. Those were my sentiments at the time and they still are.

Here's hope for those large files

Fm LDOS Support: I've come up with a new compression scheme, called SMASHing. The resulting files are very, very small. Versions of the compression scheme have been ported to many machines already. The only problem is

that I've yet to be able to extract the original files on anything.

Information on Operating Systems

Fm Adam Rubin: Anthony De Vito, Jr. [pointed me in this direction] The April 1986 ACM Operating Systems Review had a five-page "Guide to Operating Systems Literature" by Jeffrey A. Brumfield, "to introduce students who are beginning research on operating systems to the various sources of literature." It lists several dozen periodicals, conference proceedings and papers, texts, and more.

Reprints of past issues of TMQ

Fm Gary Phillips: Now that you have an in-house printing facility, Roy, any chance of getting some reprints of the first issue of TMQ? (at a premium price, I'm sure, but is it feasible at all?)

Fm MISOSYS, Inc: Brenda and I discussed the idea at length and reached a consensus of opinion. Here's just some more work for the copier machine. We ordered a new box of 3/8" GBC bindings which are just the right size to bind a TMQ. Having those now on hand, we can offer reprints of TMQ effectively immediately.

Kudos, we love them...

Fm Bob Haynes: I want to take a moment to express my appreciation for your continued support of the Model 4/4P/4D series. I'm especially pleased at the continuation of TMQ (quality, no question about it) and your active presence on the LDOS Forum. I'm under no illusions as to how long this support can continue, considering the current market, and I'm actually glad to see you pursuing alternative MSDOS activity. Hopefully we users can look forward to seeing a healthy MISOSYS for some time to come.

I've not yet gotten involved with MSDOS work as I had planned, and probably won't for some time, as health and cash flow problems have become an issue. I must spend less and less time at my favorite activity, unfortunately. When I do get into MSDOS, MISOSYS products will be a high priority, I assure you!

I suppose all of this is a small apology for lately not having supported you (with purchases) as much as I would have liked. It just hasn't been possible. I really regret not having picked up PRO-DDT and PRO-ZGRAPH when they were available. I can't justify a MARK IV package as I already have most of those programs...

Anyway! My TMQ renewal is on its way, also an order for LB and the PRO-WAM 2.0 trade up package. BTW, what's the current charge for a disk update? My copy of MC needs a "dusting and cleaning" (somehow I feel more confident when YOU do the patching).

Best regards to you, Brenda and the kids. Have a great vacation!

Fm MISOSYS: We have revised our pricing on disk refresh service fees. Check out the rates in THE BLURB.

Fm Claude E. Hunter: I recently subscribed to THE MISOSYS QUARTERLY and received issues ii and iii. I wish I had subscribed in time to receive issue i. The publication is very useful (i.e. EXMEM in issue iii). I feel compelled to write and add my opinions on the software market for TRS-80 computers. So please bear with me.

I couldn't describe myself as a typical TRS-80 user as I don't really know what is typical for this machine. I use my system as a learning tool. I like to learn how programs work and learn various programming languages. I have no real cost effective use for my computer and therefore my software budget is limited by this. I've spent about $2300 on various software products for my TRS-80 Mod 4P. I do not use many of these programs any more. ALLWRITE (with Electric Webster and Dotwriter), PRO-DD&T, PRO-DUCE, PRO-MC, PRO-MRAS, and LS-diskDISK receive frequent use.

I got very good use from SCRIPSIT and PROFILE while I commanded at Fort Stewart, GA. The frustration caused by SUPERSCRIPSIT and SUPERSCRIPSIT Dictionary while I was in graduate school was akin to the worst of torture. I spent two marathon 24 hour sessions reconstructing graduate papers that SUPERSCRIPSIT ate. The local Radio Shack denied that there were any bugs in SUPERSCRIPSIT and choked when I asked for a refund. I found out later that the version I was sold was not even the latest version and that patches had been released.

Why did I stay with TRS-80? I was exposed to Apple while at Fort Stewart and felt that none of the business related software or the operating system could compare to the ease of use of the TRS-80. I believe Apple gained such a large share of the market for three reasons: 1) Apple appeared to be a better game machine (graphics, color), 2) Apple openly encouraged outside support, 3) many Apple dealers encouraged the distribution of boot-leg game and low-grade business software. One Apple dealer offered many free boot-leg programs if you bought your computer from him and provided the disk. With this formula, how can you lose?

Why not MS-DOS? I don't like CP/M and I feel that MS-DOS is just a bulky rework of CP/M. Some day I'll have to buy an MS-DOS computer because the buying public has made that the standard and it seems that all new software development has swung that way. But as long as the Mod 4P keeps on working, I'll continue to use it to learn about computing.

In closing, I would like to thank you for the support that you have provided me and my computer. I think your products are outstanding and you take supporting them very serious. Bugs get fixed! I can understand your reasons for trimming TRS-80 products and product development. Also, I like the updates on your family in TMQ. I wish I could spend that kind of time with my children.

Too much Compuserve???

Fm Byron P. I understand the purpose of putting CIS [Compuserve Information Service] streams in TMQ. Makes it a little harder to justify the $25 when so much of it is a review of what I've read there. Sure do miss the old LSI Journal. Worth its weight (or wait) in gold.

Fm MISOSYS, Inc To Byron P. Peebles: That's a very short-sighted remark. If you think that there is nothing in TMQ beyond the little excerpts from the forum, then I wouldn't want to take your money. You are really missing out on a bundle of excellent information. Maybe your thinking is what keeps me from uploading articles from TMQ onto our forum. Then that would give you one more excuse not to subscribe.

Fm Byron P. Peebles To MISOSYS, Inc: Nice attitude (tone), Roy. Guess I'll toss the renewal notice. Save you from worrying about uploading articles and printing this forum.

Fm MISOSYS, Inc: My tone was a direct response to your attitude. No one is forcing you to subscribe. No one is forcing you to read all of the messages on the forum. I assume you are on all the time making sure you don't miss a thing. I personally have better things to do than to spend my time explaining the benefits of TMQ. 95% of my readers can do a better job doing that. And I repeat: it is an attitude such as yours which leans me away from uploading the TMQ to the forum.

On single CPU software licenses

Fm Ray Pelzer: ALL software sold by Tandy is licensed on a "single CPU" basis, EXCEPT for the Model 2/12 Bisync 3270 & 3780 packages. They did a special re-write on the license to cover one copy on ALL CPU's "at a single installation". Probably 'cuz they were selling the dumb disk for $1000 - $1500. While I was still at Tandy we ran into a hassle with this one guy in an officious position with the Saginaw School District (near Michigan's Thumb). The guy bought one copy of the 3270 & 3780 packages to use on all the Model 2's he had at the district's DP center.

As time passed, he decreed unto himself that "single installation" referred to the mainframe which the district was using, not the building in which the Model 2's were installed. This meant that all the individual schools in the district were outfitted with a Model 2, and he just gave them a backup disk of his Bisync program!

As if that wasn't enough, we found all this out a year later, when someone across the state called about a problem with a Bisync disc. They off-handedly mentioned that it had worked fine ever since they got a copy of it at the state educators' conference a few months before. I said "Huh?", and they related that this guy had made a stack of copies to take to the conference and pass around!!! He had RE-defined his decree to the definition that "single installation" NOW meant the State Board of Education!!! Now, I'm CERTAINLY not a big one to defend Tandy, but THAT frosted me! I talked to our Regional Education Coordinator about it, and she started out by saying that "oh, teachers are especially notorious for pirating software between themselves, we just

look the other way because we're more interested in selling the hardware".

Well, when I reminded her that she was paid on a percentage of sales across the whole state, and each bisync package NOT sold by Tandy was $1000 less sales, and could amount to 500 copies EASILY, AND that was 3% of half a million bucks she was throwing out of her own pocket, THEN SHE GOT INTERESTED!

The upshot of the whole thing was that Tandy legal department was supposed to send the Saginaw guy a cease and desist letter, but I never heard the outcome. It kind of got hushed up.

Here's more on software theft...

Fm Ray Pelzer: I imagine that there will always be piracy, but the idea is to limit the blatant scumbags from doing great damage. When I was teaching for RatShack, I had a group of teachers from one school district all together in an "unofficial" Z-80 assembler course. Part of the agreement to setting up this course for them was that there would be "required" software (I think it was tape DEBUG or some silly little thing) to purchase before the course began. It turned out that one of the teachers already had a copy of it, made copies for all the others, and then nobody bought anything & I wasted 6 WEEKS of class time with them! Even during coffee breaks, one guy brought in a copy of SU and was showing the rest how to use it to bootleg disks faster. sheesh!

Growing up...

Fm Adam Rubin: Summertime... nostalgia... memories of growing up in another time... anyway, here's mine. I grew up in Wappingers Falls, NY, which is right between Poughkeepsie, which is IBM R&D, and East Fishkill, which is IBM manufacturing. (I originally wrote this for another conference, but thought it might be of some slight interest here.)

Growing up in an IBM community means: Anything IBM does makes the front page of the local paper since it affects just about everyone. There's no rush hour, just "IBM traffic". School holidays are scheduled to coincide with IBM's. No one asks you where your father works 'cause it's obvious. (Mine didn't, by the way.) Your mother hangs out laundry and gets asked, "Your husband wears COLORED shirts to work?" You never have a best friend for more than four years, 'cause his/her father gets transferred... again! Your neighborhood always has at least three "For Sale" signs, for the same reason. The "Quarter Century Club" is one of the area's major social groups.

Now that's a bug!

Fm Scot Silverstein, MD: When I bought a Model 4 last year on sale, I looked inside and DID get looked at back. A big fat BUG was inside the computer!!! (And not a software bug.) A spider of some sort. I was upset and returned it, with the explanation that "it had a bug." I traded it for a Tandy 1000, which fortunately had been de-bugged at the factory.

The last word on Model I LDOS 5.3

Fm Daniel L. Srebnick: Keep up the good work, and I am one who would also be interested in a model 1 LDOS update. Any point in me sending a prepaid on that or is it a dead issue at this point?

Fm MISOSYS, Inc: I can't say any more about a Model I version of 5.3 other than what I stated in TMQ. It is a dead issue.

Here's some feedback from our renewal mailing

From Larry Schultz: Please cancel subscription. I'm in MS-DOS now.

From Leroy Gieseke: Please cancel - magazine is of no use to people who are users but know nothing about programming.

From Dennis Linder: Recently I purchased a new computer, this being a clone of the IBM AT system. I do not believe that I will benefit any longer from your services and would like to be removed from the mailing list.

I would like to thank all of the customer service people who have over the last 2 years been more than helpful when I had a problem. The level of service and the expert help given over the phone were great and also the patience with a non-programmer was special. So I say again, thanks and thanks again.

Fm Winfield Smith: I see that it's renewal time for THE MISOSYS QUARTERLY. As it coincides with my decision to switch over to the Macintosh, I think it only right to let you know that I won't be looking for new things in the TRS-80 domain any more. I let my 80-Micro subscription lapse in 1986, and now I'm cutting the last threads.

The nice looking output available from the Mac, especially with the Pagemaker and Laserwriter available for peanuts at the nearby Kinko's Copy Center, made the difference. These outweighed both the fact that my son just got a PC clone with which I could easily exchange files and software, and the stomach-turning cutesiness of most Apple/Macintosh advertising and packaging. I wrestled hard at length, but typography and layout won, with some assist from the desktop environment's superiority to that of MS-DOS.

So it's goodbye and thanks for all the support. My best to you and your family. [editor's note: sounds like a Hitchhiker's statement, "goodbye and thanks for all the fish..."]

Fm David S. Marans: Even though I am not a system programmer and I don't know my POP from my PUSH, and I unloaded my Model 4 about 9 months ago, I have still decided to re-subscribe to TMQ for two reasons: (a) All the MISOSYS and Logical Systems products I have ever owned have been excellent and I hope to acquire some of your MS-DOS programs in the future. So your product updates are valuable; and (b) I like the gossip.

Fm Richard Guerin: Thanks for the last years TMQ. Sign me up for another.

Do you suppose you could ever get, and publish in TMQ, the FELs put together by Jim Kyle for PRO-PaDS? I know they were promised a long time ago in the old DISK NOTES and you probably forgot about them. I'd still be interested in seeing them. And while on the subject of old business, how about Scott Loomer's program to use the alternate memory as a plotter? Old stuff that would make interesting reading.

Fm MISOSYS: I doubt at this point that I would burden TMQ with Kyle's FELs because they just wouldn't serve too many readers (we didn't sell that many copies of PaDS which is why it was never updated to version 2!). On the other hand, perhaps I can locate the files and put them on a DISK NOTES. The same thing may be true of Loomer's work. Anyone else out there interested in that kind of stuff?

Fm John Coyne: I have recently installed a XLR8 board into my Model 4P, and have been suitably impressed by its performance; however, it has created a problem or two. My 4P will no longer boot in Model III mode. That in itself is no longer a problem, I now use a small loader routine that will load MODELA/III into high memory, switch the hardware to Model III mode, and then transfer the ROM image into its normal running position and run. Everything is fine when running in system mode, but trying to run BASIC is a problem. BASIC appears to load, but anything you try to do from BASIC gets a syntax error. The only way out is a re-boot.

I have disassembled the boot ROM to find out why it would not boot in Model III mode, and from what I have learned managed to write the loader and get a smooth transition from model 4 mode to model III mode, but to try and dig around in the system to find out why BASIC will not run is not something I would relish. I hope you might be able to shed some light on the problem.

In the fall issue of THE MISOSYS QUARTERLY, you gave the XLR8 board a brief review and indicated that a full review would follow in the future, and also you may find the time to try and reduce the amount of low memory space the XLR8 system needs. I appreciate you have little time to spare on non-revenue earning work, however, it is your dedication in supporting your product and the Model I, III, and 4 in more than just a commercial interest that earns you my support. I am very much looking forward to reading your full review of the XLR8 board.

Fm MISOSYS: You may wish to check out Gary Phillips article in this issue. It describes a procedure to cleanly get into Model III mode on a Model 4P coupled with an XLR8 board. We also have an article on the HD64180 by David Hall. With some encouragement, he will continue to address his experiences with that chip and the XLR8 board. As for ourselves, we just don't have the time to spend on that topic. We're burdened keeping up our product development, product sales, and TMQ. Our revenues have never reached the point where we could justify more hands in the pie. And it will be quite a few years before Stacey and Stefanie can help out. Our bottom line goal is to ensure a future in this business; thus we have to limit our activities which produce no

revenue. If we crap out, a lot of folks will grieve our passing.

Fm Adam Rubin: As you've probably noticed, I'm renewing my subscription to THE MISOSYS QUARTERLY. I'm finding it very informative, and I'm looking forward to another four issues. Once we're on the subject of the QUARTERLY, I thought you might be interested in a few comments from a reader. Obviously, my overall opinion is best shown by the enclosed renewal. However, there are a few things that I think would make it a better magazine for me.

Personally, I'd like to see a bit less of the content taken directly from the Compuserve forum, and a bit more of things that I haven't seen before. Otherwise, I seem to end up paying for some of the material twice.

Would it be possible to have the section headings ("Letters to the Editor", "Programmer's Corner", etc.) somewhat larger, say perhaps two columns by two inches? This would separate the sections quite a bit more clearly than the currently easily overlooked headings, and make the magazine seem a bit less like a single 100-page article.

My last point is something you've probably already noticed. In the latest issue (I.iv), when I first read it I said "Hmm... somehow it doesn't look quite as good as it used to." After a while, I realized that the lines of text weren't quite even; some letters were higher or lower than the rest of the line. This doesn't affect the content, obviously, but it does detract a bit from the overall appearance. Incidentally, I like the use of a different color for each cover, and the information on the spine is a welcome addition. I noticed a cover price on the latest issue too; I gather its distribution is increasing.

Fm MISOSYS: Coming from you, Adam, I can understand why you may think TMQ includes more of our Compuserve information than it should. But I believe its because you are one of about a dozen extremely active users of our forum. On the other hand, the overwhelming majority of TMQ readers are either NOT forum subscribers or access the forum quite infrequently. I also recollect that the original target of TMQ was a 64-page quarterly publication. Each issue to date has exceeded that target by over 50%. I think the material we extract from the forum for inclusion into TMQ is necessary because TMQ readers gain valuable ideas concerning the use of their machines. Also, some of the material I put into TMQ may look like it came from the forum when, in fact, it has not! I use the same heading structure ["Fm name..."] for letters which we receive here at MISOSYS and put into TMQ.

I also agree that larger headings may make the separation of sections more visible. I seem to be doing everything I can to add greater content; perhaps its time to cut back and give less information. Actually, I wouldn't lose too much giving up an inch or so to the section headings; there aren't that many.

Finally, we did know about the unevenness problem. We got into a pickle with our DW-II typewheels. We use Prestige-Elite 12 for TMQ. Somewhere in the printing generation, we broke a letter on the type wheel and had to resort to our backup wheel. Have you ever tried to locate a printwheel at a Radio Shack store? They no longer stock Prestige Elite, it's a special order item (as are most DW-II typewheels). Well, to say the least, the second wheel also broke a pallet. That forced me to dredge up an old wheel which had some uneven characters; that's how that came about. Actually, the last issue had another change from the previous issue. TMQ I.iii was 116 pages and I really got burnt on the postage fees. We originally developed our TMQ rates based on the projected weight of a 64-page issue. You can imagine what that 116 pager did to our foreign postage costs. The foreign rates were developed by taking the difference in cost between the bulk mail rate and the AO Air rate of an issue. Multiply that by 4, and we derived the foreign rate (same for 1st class US). Where we had originally targeted an 8-ounce mailing to Australia, for instance, at $3.95 for postage, the I.iii issue weighed over 10 oz and cost us $5.49. Both issues i and ii came in at about 9 oz to ship at $4.72. Believe me, we couldn't afford to continue to incur that kind of shipping cost. We decided to put more on a page instead of adding pages. Issue I.iii had 57 lines of text spread in two columns of 42 characters per column. This left a 1/2 inch gutter of 6 columns. That issue was shot at 100%. Issue I.iv had 60 lines of text spread in two columns of 46 characters and a 2-character gutter. Thus, we added 732 characters per page. Since the previous issue had 4788 cpp, the revised layout increased the number of characters by 15%. In other words, a 100-page issue was equal in content to a previous 115-page issue. We also had I.iv shot at 90% to fit the increased text onto the page. Now all of that made it less readable. This issue will go back a little in page con-

tent but not to that of issue I.iii. Actually, as I write this, haven't really made up my mind yet just what the page layout will be. We will be experimenting with different values and reductions until it "looks" right. We can play that game now since our copier does reductions and enlargements in step changes.

Fm Willi Wald: Keep the Compuserve chatter and Blurb IN TMQ. A lot of very useful information is in there.

Fm Danny Stewart: For the last year I've subscribed to THE MISOSYS QUARTERLY and subscribed to the LSI JOURNAL for two or three years before that. I can make a computer do most anything in BASIC, but I never learned further than the few application programs I use and how to upgrade and patch them. All of this led me to become a dealer for the Company which makes the computers we've all learned to love, honor, cherish, obey, and curse. [editor's note: is that Tandy or IBM?]

With the changes in the market, I've been forced to abandon the computer I grew up with (Models III & 4) and move on to (boos and hisses) the MS-DOS machines. That's what sells now. I've now moved on to the world of dBASE III Plus, QuickBasic, and WORD.

While your publication is fantastic for (this is intended as a compliment) byte-heads, it usually leaves me behind. I've gained some insight, but I can see in the new world, it's not going to be of tremendous use to me. I know your magazine must count on the support from subscriptions from people. But as a businessman, I must fill current needs. I hope that, as a businessman, you also fill the current needs and will move on into the MS-DOS world as you've been hinting for so long - before you get left behind. Please put me on your future MS-DOS list. But I must let you remove it from the Model III/4 list when my subscription expires.

Fm George DeLuca: I am afraid that I will no longer keep my subscription to the QUARTERLY. I have finally made the move to a PC clone and MS-DOS.

I want to thank you for the wonderful variety of software that MISOSYS has made available to the TRS-80 world over the years. I wish you luck in your ongoing support of the TRSDOS world. I hope to see more of you in the MS-DOS world.

Miscellaneous subjects of importance to some

Fm: Jon A. Johnston: I wanted to express my growing appreciation for the sophistication of your products and the power of LDOS 5.x.x and LS-DOS 6.x.x. I've used TRS-80's since 1980 when I bought my first Model I, which is still crunching numbers in a very productive way. I used NEWDOS80 for years, and still prefer their enhanced file structure in BASIC, but switched over to LDOS/TRSDOS 6 when I bought a 4P three years ago. What a difference in the professional approach and openness with which you deal with your users. It's too bad that the computing world has gone MS-DOS, because you folks have made computing really effective for the 'personal' user.

I subscribe to the QUARTERLY and follow the Compuserve forum notes avidly. Probably will need to sign on one of these days just to keep current with what's going on in the Z80 world. I am running a clean Model I with two disk drives and a gate-array 4P with a 10 Meg hard disk at home and a Model 4 with 5 Meg hard disk at work. All of the machines have worked flawlessly for the time that I have owned them and I have a hard time understanding the complaints leveled against Tandy products.

With the assistance of THE PROGRAMMER'S GUIDE TO TRSDOS 6, I have become quite familiar with device driver programming. The enormous driver shipped with Software Support's 10 meg hard disk would not allow me to run several applications programs, most notably PROFILE 4+. Had to completely rewrite the driver to reduce it to the least code possible. Ended up splitting the driver between low memory and high memory with enough room left to run a ram disk also. The 'Guide' gave clear instructions on how to hook the driver initialization routine into the @ICNFG vector. Now the whole thing is SYSGENed and I boot directly to the hard disk.

In January, I installed the XLR8 board on the 4P. What a screamer! The board works flawlessly, and the additional .25 Meg of RAM makes a 'memdisk' a real asset. I had to rewrite the FIXBANK driver in much the same way as the hard disk driver to shoe-horn it into memory. Ended up @ICNFGing the initialization of FIXBANK also. I'm looking forward to your comments on your use of the XLR8 in coming issues of the QUARTERLY. My challenge, now, is to incorporate bank switching into the applications programs I write to make full use of the board. Would look forward to inter-

changes with you or any other users of the XLR8.

My one regret with XLR8 is that I have currently lost use of the Model III mode on the 4P, so I can't check out the new features of LDOS 5.3. I keep getting 'lost data' errors on BOOTUP when reading the `MODELA/III' ROM image into memory. I suspect disk read timing errors due to the slow start speed of the XLR8. Haven't had time to look at the code to see if something can be done. I can't imagine that the board won't run in Model III mode, and would appreciate input from anyone who has gotten it to work.

We have an old Model II running under LS-DOS 6.2 and it's a new machine. Had to patch some of the software to accommodate the different keyboard and screen but it works fine. Saw a note on running PROFILE+ on the Model II. It works fine with minimal changes. Will supply patches to anyone who wants them if they will send a SASE [10231 Monterey Cir., Northglenn, CO 80221].

Fm MISOSYS, Inc: Sounds like you have been busy. I know a number of folks out there would be interested in what you have done with both Software Support's disk driver and the XLR8's FIXBANK module. If you care to make it public, perhaps an input to THE MISOSYS QUARTERLY would be in order. Also, you will find an answer to the Model III boot problem on your 4P in TMQ. Gary Phillips worked up the procedure for an article appearing in this issue. David Hall also has some input on the HD64180 CPU chip used in the XLR8. My suite of speed tests examined in the next issue also prove the XLR8 to be a valuable Model 4 enhancement.

The TRS-80 Coverage in 80 Microcomputing

Fm Gary Phillips To MISOSYS, Inc: I received the following reply today from 80 Micro concerning my previous comments on the recent quality of the Z80 material in their mag:

> "Dear Mr. Phillips; I must agree with your assessment of the errors in Mr. Feldman's review of LDOS 5.3. "Painful" they are. All I can say is we goofed - our system failed us. We will print an update in the first available issue.
>
> As for Feedback Loop, we are aware of its shortcomings, and we are taking steps to provide better answers. Part of the problem is that readers often submit insufficient information concerning their questions, and we must do some guesswork to find an answer.
>
> Reader Forum covers a lot of territory - information for all levels of user expertise. Sometimes we will print small items that can be found in the manuals but generally aren't well-known.
>
> I don't look forward to letters of complaint, but I am glad that you took the time to write. We could all use a good kick in the pants every now and then. Thanks.
>
> Sincerely, Michael E. Nadeau Executive Editor"

Fm MISOSYS: Perhaps a useful recommendation would be to throw a few of those questions here for our "experts'" opinions and answers.

Fm Jim Gaffney: Does anyone else get the feeling that I do -- that maybe that "SECOND CLASS" on the plastic label that 80 Micro comes in refers to the publication rather than the postal rates?

It's kind of sad to look on the shelf and see the issues from January 1980 grow into a fine publication and then dwindle back down to a MS-DOS mag imitator -- and a poor one at that. I don't know how much longer I'll keep renewing my subscription. One thing for sure - renewals are on a year-to-year basis now rather than the 2 to 3 years at a time that I used to do!

Fm Gary Phillips To Roy Soltoff: I notice that in their retrospective of the TRS-80's first ten years, 80-Micro credits you and Bill Schroeder with authorship of NEWDOS! I knew you were a real producer, Roy, but I didn't realize you were behind that, too... (or would you rather not admit it?)

Fm MISOSYS: I wasn't behind that. Neither was Bill. That's just one more stupidly inaccurate remark made in 80 MICRO which reflects on the quality of their editorial staff. I got so p*ssed at that one, I haven't yet calmed down enough to write a "reasonable" letter to them to set them straight. Where they come off with remarks like that, I'll never know. I thought editors were supposed to ensure the accuracy of their articles. They even interviewed me!

Fm Kevin R. Parris To MISOSYS, Inc: Have you seen the September 1987 issue of "80 Micro", with your letter (on the next-to-last page) complaining about the review by Jack Feldman of LDOS 5.3? They include a full column length rebuttal from Feldman to your complaints. To me, the combined - effect of the original review, and this response to your complaints, give the effect of making Jack Feldman look like a buffoon. By what criteria does he qualify as a suitable person to evaluate an operating system? Is there no one among the magazine staff capable of recognizing the inaccuracies in the material they choose to print? He INSISTS that 5.3 will mess up data diskettes created by an earlier release (or, at best cause endless confusion to the user), and even suggests that on a single disk one might wind up with directory entries written in various formats! It is regrettable that the only commercial publication available to advertise your products so blatantly persists in harassing one of its advertisers this way. At least, the "editor" could insert a note pointing out that Mr. Feldman has his facts confused.

Fm MISOSYS: I read his rebuttal - if you could call it that. At this point, my perception is that 80 Micro is so far removed from the word "quality", that I can no longer recommend it to anyone. Not only did Feldman NOT respond to the points I raised, he took the opportunity to comment on things he didn't even put into the review. Rather than us going one more round, I think his words and tone demonstrated our position.

Fm Bryan Headley To Kevin R. Parris: The only thing REMOTELY like that is what I sometimes experience, between MaxDos 6.2 and 5.3. I create a disk with 5.3 (data). Then, later on, I boot with 6.2. I copy some files over from the 6.2 disk to the 5.3 data disk. I later boot with 5.3, and guess what? some of the files aren't time/date stamped right! Well, horrors, get out DATECONV on them again! Hardly a bug, just an exploit of someone who DOESN'T QUITE UNDERSTAND... I imagine this is what Mr. Feldman is trying to relate.

Fm Bryan Headley To jeff brenton: Welcome to the club. I haven't read the magazine for at least two years. Every time I do though, I go to the "Advice" column and read all kinds of outrageous answers to 'simple' questions.

Yeah. You can't interchange disks about. But, if you do, DATECONV the disk again, for the new files. Maybe Roy & Bill should have said this. But just maybe a dim light should go off by itself. Poor Roy, though. People tend to believe the first iteration of the story, and view the publisher's response as fact. Not fair.

Fm John Garner To Bryan Headley: Well, what can you expect from a magazine which reports that either Roy or Bill Schroeder (I couldn't tell from their grammar whether they meant Roy or Bill) was the author of NEWDOS!

Fm Bryan Headley To John Garner: It would be one thing if they had said Randy Cook - his name pops up in the old pre-"80" version of NewDos (NewDos Plus? NewDos 2? Whatever it was called).

Fm jeff brenton To MISOSYS, Inc: I heard from someone else who knows Jack Feldman today - it seems that Jack is adamant that Multidos is the ONLY usable DOS for TRS-80's. all the rest (and LDOS is at the top of his list in this regard) are only good for programmers, and horrible for "users".

I guess this is because Multidos is the only one that will put up with his practice of sticking disks into the machine without regard for what DOS created the disk!

Fm MISOSYS: You know, it used to be that NEWDOS80 was considered a "programmer's DOS" and LDOS was considered a "user's DOS". I'm not sure what makes one over the other. If no one programmed under any DOS, then there wouldn't be any programs. A DOS is like a car; every one has their favorite, and most are equivalent. It's like religion, also; you can't argue it because they are personal choices.

Fm Pete Granzeau To Roy Soltoff: NewDOS 80 was a "Programmer's DOS" because one had to be a programmer to figure out those bodacious cryptic command lines.

Fm Louis Self To LDOS Support: Congratulations on maintaining such good communications and help for your users through the forum. I don't write much but do get much good information by reading questions and answers from others. Saves me embarrassing myself by asking the dumb questions that really need to be asked.

Now I am confused about which DOS will read disks from which DOS. I am getting different messages from 80-Micro, the documentation that came out with the DOSes and here on line. This

is the equipment I have. I interchange data disks extensively between all three systems:

MACHINE	DOS
Mod 1	LDOS 5.1
MOD 4 in 3 mode	LDOS 5.3.0
MOD 4 in 4 mode	LS-DOS 6.3.0

Please don't tell me to get rid of my Mod 1, as it fills a need and gets much use.

I feel a chart cumin' on that might help in the explanation. Maybe you can think of a better way to explain it or would like to include other DOS versions.

For example LDOS 5.1 can read, write, use and use dates from a disks from itself, LDOS 5.1. It can also use a disk formatted under itself, LDOS 5.1). [editor's note: modified chart appears later]

It would be a big help if you would download this chart, revise it as necessary (in upper case) and send it back in your reply to this message. Also If there is nothing like this chart in your DL you would be welcome to use this one if it would be helpful to others:

One more thing. Under which DOS should one format disks to make them most transportable to and compatible with all 3 DOSes? Whew! What a lot to ask! Thanks for your help.

Fm MISOSYS, Inc: Writing to a 5.3 or 6.3 disk from 5.1 or 6.2 (or earlier) will introduce an erroneous date in the directory of the file being written. The file is still perfectly usable under 5.3 and 6.3, but it has a wrong date (and time).

Fm Louis Self To MISOSYS, Inc: Thanks for your reply, Roy. I am to my neck in this and my brain is beginning to go BONGLE, BONGLE, but please stick with me just a little more. I have changed the chart by: (1) Adding TRSDOS 6.2 (just cuz); (2) Modifying it according to your suggestions; (3) Adding upper case in areas where I am uncertain (check, please); (4) I understand from you and the DOS instructions, that a disk from any DOS in the chart can be used with any other (with some restrictions on date, time and password). Therefore I have added F's to all locations. (5) Adding "t" for time/date in the directory; (6) Adding the NOTE; (7) Touched it up cosmetically.

Now just 2 more questions: (1) For disks to use under 5.3 & 6.3 would it be best to format them under 6.3 (rather than 5.3)? (2) The blue updating instructions for LDOS-5.3 on pg. 2 say, "Also, once you have updated a disk with the DATECONV/CMD program, do not write to that disk using an earlier release of LDOS." Is that statement somewhat erroneous?

Fm MISOSYS, Inc: You can format a disk under either 5.3 or 6.3 to use under either. That's for use as a DATA disk. It is best to FORMAT a disk with the DOS in question if you are going to make it a SYSTEM disk. But then, don't you think that is self evident? On the other hand, QFB or DISKCOPY can be used easily to clone a system disk of either DOS. The "blue" instructions with LDOS say not to write to a disk from pre-5.3 (or pre-6.3) once date converted (or once formatted with either 5.3 or 6.3) because of what I have already stated. The file written to will lose the correct date/time when accessed again by 5.3/6.3.

Your chart would be a whole lot easier to look at by narrowing it to 2 rows and 2 columns. For the purposes of file access, 5.3 and 6.3 behave identically. Thus, you can forget about referencing 5.3 and 6.3 distinctly and refer to it as "x.3". Next, you can refer to LDOS 5.0.x, LDOS 5.1.0, 5.1.1, 5.1.2, 5.1.3, and 5.1.4 (any 5.1.x) as well as TRSDOS 6.0, 6.1, and 6.2 as the same grouping. As far as access goes, they would be equivalent. Call them "pre-x.3". Why do you think the LDOS release was called 5.3 instead of 5.2? It was to avoid confusion with TRSDOS 6.2 and correlate the nomenclature better with 6.3. Here's my revision to your chart:

DOS COMPATIBILITY		
SYSTEM DOS \/	DISKETTE DOS Pre x.3	x.3
Pre x.3	r w d	r *w d
x.3	r w d	r w d t

r = Can read data from the disk
w = Can write data to the disk
*w = Can write data to the disk but date & time will be incorrect in directory
d = DATES CORRECT WITHIN THEIR RANGE
t = Time/date stamp will be correct
NOTE: AVOID PASSWORDS WHEN SWITCHING DISKS BETWEEN OLD & NEW DOSs.

Fm Bob Schindler To Pete Granzeau: I don't know why everybody has their bowels in an uproar over this anyway. It seems (in most cases anyway, thank you very much Hardin!) that reviews of software, hardware, etc. come from

the same cauldron as Siskel & Ebert at the Movies. Anyone with an opinion, however off-base it is, can be a reviewer and a CRITIC! The proof is in the pudding. Anyone who wants to know about a particular product, be it software or 'Crocodile' Dundee, needs to (1) ask someone who knows, or (2) take his/her lumps for being adventurous and blowing it, or (3) be adventurous and successful. I rarely take any so-called reviewer seriously. Besides, the problems voiced in the review, at least as far as my experience extends, seem to come from Vulcan or some such. 'Nuff said.

Fm MISOSYS: You would be surprised how may folks will absolutely make their buying decision on a review. We just got an order today for a 5.3 upgrade kit from someone who said he didn't order it before because of Feldman's review. When he read my rebuttal, his mind was changed. Unfortunately, once something appears in print, its very difficult to undo the damage that was done. Now that 80 Micro wrote that I and Bill Schroeder wrote the "unauthorized NEWDOS", how do you think I can undo that damaging remark? Certainly anyone who reads the mail on this forum knows that Bill and I had absolutely nothing to do with NEWDOS. I mean don't 80 writers read Decker?

Fm Ray Pelzer To MISOSYS, Inc: hm... removed from the word "quality"? Or perhaps, to paraphrase the old chestnut: "80 Micro is for people who can't handle reality."

Fm H. Brothers To Bob Schindler: It has always seemed to me that there are two parts to a review. First, there are the facts (you don't want me to tell you that a graphics program is really a word processor, for example). If a reviewer messes up the facts, he/she has screwed up the review.

Second, and more important to me when I'm reading a review, is the opinion. Is a product good at doing what it does? All reviews SHOULD include a healthy dose of opinion I think. The opinion should be "informed" of course, which is why those of us who have written a lot of reviews are asked to write more -- we're supposed to have some background in the type of products we're reviewing. If you read a reviewer (or listen to a movie critic) long enough, you get a feeling for how well your opinions and the reviewer's coincide.

Suppliers are, of course, happy when a reviewer likes their product and unhappy with a negative review. However, if the negativity is based on opinion and the reasons for that opinion are explained, I don't think the supplier has any basis for complaint except hurt feelings (and perhaps profit). On the other hand, if the negativity is based on factual error, or if there are any substantial errors in a review, the supplier has a good cause for complaint.

When I'm criticized for errors, I try to correct them. When I'm criticized for my opinions, I tend to become defensive. Unfortunately, I know some reviewers who become defensive in the face of any criticism.

I find reviews much more difficult to write than tutorials. First, the responsibility to both reader and supplier is greater and, second, deadline pressure almost always means that I don't get to use a product as long as I would like before writing the review. It's been almost 18 months since a supplier last threatened my with bodily harm for my opinions, but I'm sure it will happen again some time.

jeff brenton To MISOSYS, Inc: Well, I consider LDOS to be a "programmer's DOS" because EVERYTHING WAS DOCUMENTED. This, of course makes the manual very thick, which literally scares a lot of "users". "I've got to read THAT before I can use this?" is stated. Isn't it strange that they're intimidated by the size of the manual, when they never read them anyway? B-)

NEWDOS was considered a "programmer's DOS" because of what they did to BASIC. but it was a "user's DOS" because it was the first one to crash less often than TRSDOS!

Fm Bob Schindler: I concur fully with informed opinions. I take EVERY review I read with a grain of salt, so to speak. I also agree with Hardin's comment about being familiar with a reviewer's style and tastes. There is a movie reviewer on local TV here in Cincinnati whose word I will take on a film, play etc. Anyone else's is horse puckey to me. One review I saw even had the audacity to say that the Star Wars movies were total trash from square 1. Granted they aren't great literature, but shucks, for plain FUN they are great.

Fm Ray Pelzer To MISOSYS, Inc: Would a court look on that NEWDOS comment as libelous, or just as misfeasance?

Fm MISOSYS: To tell you the truth, I have given the litigation issue some concrete thought. That statement was KNOWINGLY false

because the general public knowledge was such that NEWDOS was known to come from Apparat. Now the bigger issue is was I injured by that remark and was my company injured by that remark. I would love to bring suit to put that rag into the position of knowing that they best get their act together because we won't stand for those kinds of statements. I believe that I have already lost sales due to that statement; proof of that is another issue.

Fm Bob Schindler To MISOSYS, Inc: It's really a shame that there isn't a Constitution for reviewers. It seems (in my li'l ol' punkin head idea now) that a review should be conducted as Hardin described his methodology, with one minor alteration: The reviewer's opinions should be (if possible) heavily backed up with supporting facts. Such - and - such procedure on a word processor is cumbersome. WHY? It is this type of information that makes a personal opinion worth a d--n. Simply stating that something is less than desirable without saying why is like telling the doctor you have a pain (grin), but not where it is. I really do understand your position; my last job went away because of a similar situation.

Fm Phil Oliver To Bob Schindler: People could learn a lot from the rules of evidence painstakingly created over the years in the law courts. Concepts such as hearsay and burden of proof and substantiation of opinion by facts in evidence are a few such principles.

Fm MISOSYS: About four-five years ago, I refused to renew my subscription to TIME magazine because it continued to be full of advertising for cigarettes (I am a converted and aggressive non-smoker). I wrote them a letter stating the reason, and if they chose to drop their cigarette advertising, I would re-subscribe. I doubt that it affected TIME, but I sure felt better about it. Perhaps folks choosing to not subscribe to 80 Micro should write them a letter telling them why not.

Fm Ken Hipple To Bob Schindler: One of my big problems with reviews is when the reviewer makes his opinions sound like facts instead of clearly stating that what he is saying is just his opinion (no matter how 'informed' that opinion may be). I've seen more than one review say that a program was terrible and shouldn't be bought for the very reason that I like it. And then I've also seen reviews done right that say, 'I wouldn't buy this because it works the way it does but you might like it doing that'.

Fm LDOS Support To Marc Nowell: The response was unbelievable. That 80 Micro would print it, however, is believable. The sad part is, many folks will read it, and, not knowing any better, will believe it.

Fm Bob Haynes To MISOSYS, Inc: Roy, despite hating to lose the last decent Model 4 support in any magazine circulation (Hardin's column) with the notable exception of TMQ, I am forced to agree with your comments. If each Model 4 subscriber took the time to write an intelligent missive to that effect, perhaps some good might come of it. I fear, though, that there will be far fewer than needed to persuade 80-Micro's management. Oh, well, mine is but one vote, nonetheless, I SHALL cast it!

Fm Kevin R. Parris To MISOSYS, Inc: I guess you are right. Trying to send in another letter complaining about Feldman's response to your first letter probably would simply generate more of the garbage we have already seen.

Fm MISOSYS: The power of the press is such that the press always has the last word - until they get taken to court and lose.

Fm Marc Nowell To MISOSYS, Inc: I agree with Bob regarding that the only thing worth getting 80-Micro for is Hardin's stuff. I, too, will send a letter. Now, if Hardin's column was in TMQ instead of 80-Micro, then, yeah, that's it, we wouldn't NEED to look at 80-Micro anymore, yeah, that's the ticket!

Fm Phil Oliver To MISOSYS, Inc: As I mentioned before, I stopped subscribing to 80-Micro years ago after it really started appearing to be fading away. Now that Tandy is making only IBM PC clones (except for color computers, or have they stopped making those too?), 80-Micro has severe competition from the magazines who have made it their business to cover that market for much longer than they have. The editorial quality of the articles and reviews in PC Magazine, PC World, etc. far exceed 80-Micro. I can't think of any better reason to drop a subscription to the magazine.

Fm MISOSYS: It seems that everyone who has responded to me recently gave up their subscription long ago. My hunch is that CW Communications started PC Resource as a graceful way to drop 80 Micro. At this point, there is absolutely no need for a vendor-specific magazine in the MS-DOS word. Tandy's MS-DOS based computers are clones or near clones. Why have a magazine devoted to them?

LDOS Model I/III Information

statement (nnnnn represents some decimal value) which clears string space for the program. What you should do is to reduce the value by 435. Then your program should run. If you do not wish to do that procedure, then the alternative is to copy the LBASIC/CMD file and its overlays from the 5.1.4 disk over to the 5.3 disk and use LBASIC under 5.3. A "BACKUP LBASIC:s :d" command will suffice for this. The 5.1.4 LBASIC facility still works under 5.3; thus, you will have all of the 5.3 added DOS features without added BASIC enhancements.

Extra paging out of SCRIPSIT has absolutely nothing to do with the DOS. I will assume, of course, that you do NOT have the PR/FLT forms filter installed. That should not be installed if you are going to use SCRIPSIT as SS performs its own paging. Double check your local procedures with the SS problem - it really cannot be the DOS.

Time Manager, Project Manager, and Scripsit

Fm John P. Caire, Jr: I purchased your LDOS 5.3. It is installed on a Model 4P with the Radio Shack 5MB hard drive. This is not a new installation. I have operated this combination for three years utilizing the original LDOS which came with the Profile III Plus hard drive package.

I had also been running the following programs which are not operating properly on the hard drive with LDOS 5.3: Time Manager, Project Manager, and Scripsit. Both Time Manager and Project Manager operate with LDOS 5.3 on floppy disk. Neither will operate on the hard drive (they did before the conversion). I get not enough room and the programs will not work; both drive 0 and 1 have more than 200K free.

I am experiencing problems with Scripsit. I get multiple page feeds at the end of a document. Usually three.

Fm MISOSYS, Inc: I may have some easy answers for you. TIME MANAGER and PROJECT MANAGER sound like they are written in BASIC. Your "not enough room" error message explanation sounds like a BASIC error indicative of MEMORY space, not DISK space. I suspect that those programs used all of available memory. The BASIC/CMD provided with LDOS 5.3.0 is 435 bytes larger than the LBASIC/CMD supplied with LDOS 5.1.4. Somewhere in the programs you were trying to use you may find a CLEAR nnnnn

Updating the CMD"V" patch for LDOS 5.1.4

Fm Larry L Hildebrand: I hope either you or someone who reads your QUARTERLY can help me. In the LDOS QUARTERLY (Vol 2, number 3), an article "Active Variable Dump for LBASIC" was published for Model I LBASIC 5.1.3. When I upgraded to LDOS 5.1.4, I converted all my system diskettes to 5.1.4 only to find that LBASIC had been changed so the LBASOV4/FIX file doesn't work. Since I no longer have a copy of LBASIC 5.1.3 I can't compare LBASIC 5.1.4 to find where the new addresses are. HELP! If I had a hex printout of LBASIC 5.1.3 or a file, I may be able to find the new addresses, or if you or someone could revise the LBASOV4/FIX file for LBASIC 5.1.4 Model I, I would be forever in their debt. I'm lost without my CMD"V". Look forward to your reply or in THE MISOSYS QUARTERLY.

Fm MISOSYS, Inc: I think you will find these changes to work. There was a patch added to LBASIC 5.1.4 which went into the same spot.

```
. LBASIC 5.1.4 patch for CMD"V"
. Re: LDOS QUARTERLY, Vol 2, Num 3, p16
X'5464'=C3 BE 64
X'64BE'=FE 58 CA 68 54 04 04 FE 56 CA 68 54
C3 D7 54
X'5803'=C3 CD 64
X'64CD'=04 04 FE 56 CA 68 54 C3 4A 1E 00
X'53CE'=64 D7
. Eop
```

Installing the Hardware Interface KI4 driver

Fm James Peterson: Here's a note re the LDOS 5.3 Model 4 Interface Kit. I wrote in regard to difficulty in installing KI4/DVR. The patch you returned had been installed. The problem was caused by my using the "USING" option. When I left it out of the statement,

```
SET *KI KI TO KI4/DVR [USING] (TYPE,JKL)
```

it worked fine.

Fm MISOSYS, Inc: Sorry, that was our flub. The "[USING]" should never have gotten into the documentation. The preposition "USING", as well as the other three: "TO", "ON", and "OVER", are always permissible preceding a file specification; however, none of them are permissible anywhere else unless the program parsing the command line specifically checks for them. These four prepositions are permissible in front of filespecs because the @FSPEC service call will skip over and thusly ignore these four strings. That's why you can't have a filespec the same as one of those four names unless you don't use the DOS @FSPEC service call on it.

Making a REAL 2-sided BOOT disk

Fm Jim Hawes: Now here's more on LDOS 5.3. First a point of trivia: item (1) [Here's a] BOOT/FIX for LDOS 5.3 to get correct system name on boot sector 3

```
PATCH BOOT/SYS.SYSTEM
 (D03,D6="30":F03,D6="13")
```

Item (2) There is an infinite loop in DIR which occurs if you make a parameter, date format, or sort buffer size error from a JCL. It looks like you omitted an "@". The jump at 5418H should go to 4030H not 5A28H which I'll bet was labeled "ABORT". I guess the patch should be,

```
PATCH SYS6/SYS.SYSTEM
 (D07,0D=30 40:F07,0D=28 5A)
```

Item (3) This refers to a note on RESET appearing on page 25 of TMQ I.iii identified as coming from LDOS Support. Since there are a lot of double sided drives out there, I think the record should be set straight. If you wish to print any of this, feel free to edit as you see fit. As I see it, the recipe for making a double sided system disk should be:

```
FORMAT :1 (Q=N,NAME="LDOS-530",SIDES=2)
BACKUP SYS0:0 :1 (S,I)
BACKUP :0 :1 (S,I)
PATCH BOOT/SYS.SYSTEM:1 BOOT2S
Finagle with boot sector 3 (optional)
```

Here's BOOT2S/FIX which corrects the DCT images on BOOT sector 2.

```
. BOOT2S/FIX for LDOS 5.3 - Corrects
. boot sector DCT images of a 2-sided
. system disk created from a 1-sided system
D02,24=61:F02,24=41:D02,74=61:F02,74=41
. Eop
```

I do not know the function of the DCT$ image at 02,20 so you can help out. Is it just a backup or does it get used? I understand the use of 02,7A - 02,BF by SYS0/SYS and the control of bit-5 of DCT$+4 by SYS0 as it gets the correct bit value from DIR sector 00,CD bit-5. The problem with RESET (entered as a typed command and not the orange button) is of course that global reset does copy BOOT sector 02,70 - 02,BF to the DCT$ and if bit 5 of 02,74 does not reflect the true state of SIDES, well there goes half of your disk operating system.

As to boot sector 3 which FORMAT gives an image of the original GAT sector, I do not know how necessary it is to correct it for the change in allocation (mostly changing a bunch of X'F8's to X'C0's and of course the X'CD' byte). There are no doubt several ways to get it done. Here's what I did:

FORMAT two identical disks at step 1 above. Call them A and B. Backup files to A only! Then DEBUG (E) <BRK> and read sector 3 with disk B in drive :1. Type, 1,0,3,r,6000,1 <ENTER>. Switch drive 1 to disk A and write the sector back with 1,0,3,w,6000,1 <ENTER>. Now exit from debug via the <0> command. Type DEBUG (E,N) <ENTER> to remove the debugger from memory and apply BOOT2S/FIX to get the finished product. To get further 2-sided backups, LOG :0 and switch to the good 2-sided disk in drive :0. Now QFB :0 :1 with the good 2-sided disk in drive :0 produces a perfect 2-sided disk which can be booted, RESET, etc. BOOT2S/FIX is not needed again.

As to "don't global reset", I take exception! Obviously if your modem is taking a memo from Col. North or your pacemaker is being run from a high memory driver, I do not recommend its use. When you are finished with a task and want to start another that requires a different configuration there are 4 approaches. (1)

turn off the computer and restart 30 seconds later (some delay is necessary if you want everything clean). (2) hit the orange button to force a re-boot. (3) type BOOT <ENTER>. (4) type RESET <ENTER>. BOOT or the orange button may be required if your computer has crashed or you have made some curious alteration below 4E00H, but RESET is a little more mild, takes less disk access and may have some positive advantages over the orange button. I have two model 4s. With my older machine, the orange button is "clean" in the sense that it does not change bytes above 5200H. The 4D has what I call a "dirty" button in that it causes considerable spray of interesting bytes to appear throughout memory. From time to time, I need to start fresh but I want to save the actual code in high memory and the alternate banks. Global RESET is what is called for. Every thing does have its proper use.

I started looking at the code associated with DRIVE and its secondary parameters - CYL, DELAY, ENABLE, DISABLE, STEP, DRIVER, and WP. What the 513 manual does not say is that: SYSTEM (parm, parm, parm), where (parm) refers to ALL of the above secondary parameters, is global for all eight drives. SYSTEM (DRIVER="file") is of course a no no.

Fm MISOSYS, Inc to Jim Hawes: If you want to be a stickler for detail, you can change the packname noted in the GAT image of track 0, sector 3; of course, you would want to change the date too, wouldn't you? Actually, that GAT image serves no purpose whatsoever in the architecture of the system. It happens to be a copy of the GAT when the disk was formatted. A BACKUP to that disk doesn't change that sector, so it really represents the original format to that diskette. Duplications of that diskette will make clones of the disk, regardless of what operating system version is present. Thus, it can be argumentative whether the "-513" nomenclature is incorrect or correct. Also, since that image is never used anywhere in the DOS outside of the original generation by FORMAT, it serves no purpose to alter the contents for any reason (i.e. constructing a 2-sided boot disk).

Now as far as your procedure of making a 2-sided boot disk, I really feel it is overkill. Can you imagine us trying to present that as a procedure to some user out there who doesn't know a disk from a disc? Certainly, we cannot expect that procedure to be applied by the typical user. You have already recognized the impact of a global RESET. That's why Joe makes the statement he does re recommending against it. On the other hand, if some TMQ readers are interested, I'll read over your procedure in detail and get it into that publication.

The function of the DCT$ image at 02,20 is for the Model I LX-80. The LX-80 had a special hardware trap which took over the ROM boot function and read sector 1 of track 0 instead of sector 0. One of the things I changed in the operation of LDOS was to merge the LX-80 version and RS E/I version into one disk bootable on both configurations. The Model I SYS0/SYS init code detected which E/I it was working on and captured the 02,20 DCT's for LX-80. If you go back to your Model I disassembly, I am sure you will see that.

Incidentally, the reason why some Model 4's screw up RAM when pressing the reset button is because the action disables RAM refresh, possibly just on some machines. You can sometimes avoid it by QUICKLY pressing the button and releasing it.

Lastly, let me comment on the parameters associated with the SYSTEM (DRIVE=) command. Your statement that "all of the above secondary parameters, is global" is not correct. The CYL= parm is ignored if the current drive type is 8"; the DISABLE parm is ignored unless a DRIVE= is passed; ENABLE is global; STEP is ignored if the controller is alien (which also implies HD, as alien is supposed to be flagged for HD drives); DELAY it turns out is done globally for all drive types unless STEP is entered at which point the DELAY code is bypassed if the drive type is alien (this is a bug in operation). Note that specifically, "SYSTEM (DISABLE)" is prevented by code which explicitly tests for the presence of a DRIVE= entry. The extreme variation in the other parameters gives good sense to the scope of the documentation. You really don't want to confuse too many folks.

The PATCH command's OPTION parameter

Fm Clay S. Scott: I would like to know how to prevent LDOS 5.3 from requiring a Find line for the patch command. The new spec sheet for LDOS 5.3 says the default is to not require a Find line.

When I tried to apply the patch SCR17/CTL (D02,40=28) I got a "Find line mismatch." The SCR17/CTL file was on a 5.3 version disk. However, when I booted a 5.1.4 version of LDOS

with the SCR17/CTL file on drive 1, the above patch worked. Is OFF a switch for OPTIONs?

Fm MISOSYS, Inc to Clay S. Scott: The documentation is incorrect in the default state of the switch: it's default is ON to keep it consistent with LS-DOS 6.3 (as well as all other 6.x releases of TRSDOS). If you would prefer to have the OPTION parameter default to OFF, this can be easily done via a patch to PATCH as follows:

```
PATCH PATCH.UTILITY
 (D09,3D=00 00:F09,3D=FF FF)
```

Thereafter, the option parameter will be set as to not require FIND lines. Note that you will then have to explicitly set (0) when you wish to apply a patch provided by MISOSYS as we will nearly always provide direct patches using FIND lines.

Notes on RSHARD - Hard Disk driver

Daniel L. Srebnick To MISOSYS, Inc: Roy, I have noticed that when using HDCHECK5 the sector counter does not change, while the cylinder counter does increment. Also, HDCHECK5 does not detect errors on granules that HDCHECK6 locks out. I am using a green screen model 4 (gate array version) with the clustered arrow keys, LDOS 5.3.0, SET2RAM and the BANKER. Have you heard this one before?

Fm MISOSYS: No, but if you hum a few bars, I'll check into it. Seriously, Dan, I took a look at HDCHECK5 at your suggestion. Actually, it wasn't staying on 00 sectors; when I ran it it was shifting rapidly between 0 and 1. The cylinder count was also screaming. The bug was due to register utilization in calculating the number of sectors per cylinder (actually due to some conditional code between the 6 and 5 versions). Here's a two-bit patch to correct it.

```
PATCH HDCHECK5 (D00,D6=59:F00,D6=5F)
```

I suspect that it was not catching bad sectors caught by HDCHECK6 since it was not checking the full cylinder.

Tim Clute To MISOSYS, Inc: I would like to know if the RSHARD drivers will permit a 5 Meg partitioned into 4 1.25 M logical drives to have 1k/gran or must we continue to live with the rather large grans we get at present with TRSHD6/dct?

Fm MISOSYS: The RSHARDx/DCT drivers would default to 2K grans on either 1 or 2 head partitions. This could be changed with a 2-byte patch to support 1K grans. There is a grans-per-cylinder, sectors-per-gran table in the RSHARDx/DCT driver. The table is 8 pairs of numbers corresponding to a heads-per-partition of 1-8. The tables are at 359DH and 656FH on the V6 and V5 drivers (at least on the last set of printouts which I believe are current). Present values are 3,8,7,8,15,6,,,.

Tim Clute To MISOSYS, Inc: Thanks for the info. BTW is there a setup program akin to Hardgen/bas in the package?

Fm MISOSYS: No, there is no "setup" program other than the detailed instructions we provide on the methodology of partitioning.

TRSHD3/DCT and SET2RAM - Warning!

Dave Krebs To LDOS Support: I tried to execute a JCL this evening with SET2RAM installed and it BOMBED the system. The whole screen was filled with "=" signs. My system is a 5 Meg HD with 5.3 and 6.3. The only thing in memory besides SET2RAM was TRSHD3 and KI4. The JCL was:

```
BASIC (e=n) run"gl"
```

If I executed it like this ==> do gl, the thing bombed, but if I did this ==> do = gl, everything was OK. Any thoughts?

Fm MISOSYS: Check your README/TXT file that came with the HIK to see if the SET2RAM patch has already been installed. If not, here it is.

```
PATCH SET2RAM (D00,50=00 38:F00,50=55 37)
```

If not installed, then a BASIC program using TIME$ would bomb on some versions of the ROMC but not others. Also, have you patched your TRSHD3/DCT driver per the note mentioned in the Hardware Interface Kit docs?

Dave Krebs To MISOSYS, Inc: Thanks, I'll check the README but I found the GOTCHA. I fixed TRSHD3/DCT when I got the LDOS 530 package with the patch for the "write to ROM" problem and thought about it again at work today and found that when I upgraded with the patches from Disknotes7 I somehow lost the patched version of TRSHD3. Oh well, I fixed it again and everything works as advertised!

Here's more on TED from Jim

Fm Jim Hawes: I just came across an anomaly on the TED docs. Evidently at some point after you wrote the docs, you changed the cursor control situation. The docs that I have state:

<ctrl H> same as <LEFT ARROW> key
<ctrl I> same as <RIGHT ARROW> key
<ctrl J> same as <DOWN ARROW> key
<ctrl K> same as <UP ARROW> key

When you went to Extended Cursor Mode (ECM) this became inoperative. Also should you point out to folks that you are using the ECM and that if they inadvertently hit the <CLEAR SHIFT SPACE> they will lose cursor control until they repeat the toggle? If you feel that this is even worth mentioning - (I am sure most folks could wear out their keyboards before they trip on this) - would it also be worth pointing out that the toggle for insert/overstrike (00,5A=1A) could be customized to <CTRL I>?

Here's more. This one you will have to play with, and the solution may be a code PATCH or simply "don't do it if it hurts".

If you have text in the buffer and exit, the docs say that you can re-enter with "TED *". Fine, but don't do an EOF (<CTRL F>) as your first command upon re-entry! TED appears to have lost an EOF pointer so we slip off the end of the world. Worse yet, try to file to a clean disk! You're going to get a looooooong file! The best thing I've found is to search the entire text for any string or scan from top to bottom with the <DOWN ARROW> - these update the EOF pointer and all seems well from then on.

Fm MISOSYS, Inc: You're right on both counts. The difficulties stem from inadequate documentation. Realize that TED/CMD originated from our TED/APP application that is part of our Mister ED application pac. That is a special module which works through PRO-WAM. Those applications had very tight restraints on their length being forced to fit into a 2K piece. Also, TED/APP originated on a Model 4 under DOS 6.x which doesn't have an ECM; it doesn't need it! Since I had some leeway in the keyboard code generation when I did TRSDOS 6.0, I was able to have UP ARROW generate a 0BH instead of the LEFT BRACKET, "[". That was the entire reason for ECM under LDOS - so that UP ARROW could generate something unique from "[".

The documentation for TED/CMD was the same documentation as that which I used for TED/APP - with a little editing. I did miss that bit about the ARROW keys and their CONTROL key equivalents; that's only correct for the Model 4 version.

As far as the "TED *" facility, that's another documentation blunder. Since PROWAM applications do not get invoked by a command line interface, there is no way to add a command line parameter like "*". What was done in TED/APP was to create an entirely separate application which differed from TED/APP by one byte - the byte which was the non-zero part of a "LD (HL),0" instruction. As you may be able to guess, this instruction placed a zero in the first byte of the text buffer. By eliminating that instruction, the text was left intact. Of course, the EOF pointer was incorrectly established. That's why the documentation for OOPS/APP (the equivalent of TED *) had this paragraph:

> One cautionary note. Since all of the text pointers normally established by TED will not be initialized when invoking OOPS, it will be necessary to scroll through the text until reaching its last character prior to doing any other operation. This may also be performed using NEXT PAGE.

This paragraph should have gotten into the LDOS section on TED/CMD after noting about "TED *".

Moving files from LDOS to TRSDOS 1.3

Fm Billy D. McDaniel: Does anyone know how to convert from LDOS format to TRSDOS format? TRSDOS to LDOS is no problem but the other way around has me befuddled.

Fm MISOSYS: It's actually pretty easy. Under LDOS 5.3, FORMAT a 35-track single-density diskette. Then COPY the files to that disk. Boot up your TRSDOS 1.3 and use its CONVERT utility to get the files. TRSDOS 1.3 will think that the 35SD disk is a Model I disk. The directory track must wind up on track 17 (it will invariably) because Tandy wasn't smart enough to have CONVERT look at track 0, sector 0, byte 2 for the directory track pointer. The only problem you will have is if you have a file greater than about 80K.

Fm Jim Kyle: The SD trick works for ALL of the M3/M4 DOSes (deese?); it is the only totally compatible method of moving things from one DOS to another! In extensive experimentation I found that the SD disk created by any DOS I tried (LDOS5, LDOS6, TRSDOS1.2, TRSDOS1.3, NEWDOS80, DOSPLUS, or MULTIDOS) could be read by any of the others, so long as it was created on a Model 3 or Model 4. Did not have a Model 1 to try with but would not have expected it to work with any of them due to the different [Data Address Mark] DAM. It is SO reliable, in fact, that it was the format I chose to distribute products of "the software factory" in (an idea shamelessly stolen, with ex post facto permission, from our present host Roy, who did the same thing and also provided a patch to permit Model 1 users to read the same disks!). "tsf" is now, unfortunately, essentially out of business although support is still provided (1986 total sales were <$25).

DATECONV and diskDISK '/DSK's

Fm Craig Baker: I just received my LDOS 5.3 upgrade and have several questions not covered in the documentation. First, do I use the DATECONV/CMD utility on DISKdisk files? Or do I need a different DISKdisk driver for 5.3? Also I'd like to get the extra 12 directory slots on my hard drive that 5.3 allocates on "data" disks. Does DATECONV do this automatically? Or do I have to re-format the hard-drive. Lastly, I assume all the old hard disk drivers and utilities will work under 5.3, correct?

Fm MISOSYS: DATECONV is needed on diskDISK files since diskDISK has not been converted to the x.3 dating convention [there are patches for that in this issue of TMQ]. The problem with the 12 extra directory slots is that TRSFORM would need a patch to reflect the formatted partition as a "data" drive. I don't believe that I have worked up any patch to TRSFORM3 or TRSFORM5. The easiest way to achieve the end result would be use FED to change the disk configuration bit. That's bit 7 of relative byte CD in the GAT (1st sector of the directory). Change the bit to be a "1" and you now have a data disk. Perhaps someone may have some time to look at TRSFORM3 and TRSFORM5. The formatter that's bundled with our RSHARD package sets that bit. Your existing hard disk drivers will work with 5.3.

BASIC and '/BAS" file extension

Fm Ed Boudrie: I recently installed the LDOS 5.3 upgrade to my TRS-80 Model IV with a 15 meg hard drive. However, I am having problems with getting several SYS errors in addition to some problems with my Account Payable program. For some reason, the system cannot find the files for this program even though you can see that they are on the drive through the directory. The A/P program is written in Basic. I should note that I was previously using LDOS 5.1.4 without any problems. Any suggestions?

Fm MISOSYS: BASIC programs must have the file extension "/BAS", or BASIC must be invoked with the command "BASIC (E=N)", or the program must be invoked with the command "RUN prog-name/". All of this is covered in the LDOS manual. Some of it is covered in the README/TXT file located on the LDOS 5.3 release disk referenced by the first paragraph of the update documentation. Nuff said?

Molimerx and LDOS 5.3

Fm Andre de la Fressange: I am still using LDOS 5.1.4 with my GENIE 1 and TANDY mod 3. Is it possible to have it upgraded to LDOS 5.3 through MOLIMERX in England or otherwise? One of my problems is to adapt the LDOS 5.1.4 to 3ms disk drives. Please could you advise me. Many thanks in anticipation of your reply.

Fm MISOSYS: Molimerx no longer is involved with the TRS-80 software market, to my knowledge. Molimerx was also never authorized to distribute LDOS 5.3 nor were any copies ever sold to them by MISOSYS. We are the only suppliers of LDOS 5.3. The evidence we have pinpoints the copies of LDOS 5.3 which Molimerx was (or is) selling are copies of the LDOS 5.3 upgrade kit we sold to a Canadian company. That serial number was TC0A0020 and was registered to J&J Electronics Ltd. There was never any authorization for that nor has MISOSYS received any funds for those unauthorized copies sold by Molimerx. Funny how dozens of copies of a disk we sold to a Canadian company have wound up in Great Britain. Such is life! And folks still complain about me not wasting my time on a Model I release!!!

Patches for extending the system year???

David D. Brown To MISOSYS, Inc: There is a file Y1987.TXT in DL5 concerning a patch to LDOS 5.1.4 to correct the date arithmetic for a new range of 8 years. Has something been resolved about this? I am a registered owner of LDOS, but never received any information about an upgrade to handle the new dates. Did I miss something?

Fm MISOSYS: I know nothing about that patch. MISOSYS has never provided any patch associated with any extension of date. We will (may) do that for Model I but not Model III. MISOSYS has been advertising LDOS 5.3 for the Model III since January. It has been discussed on our forum since November of 1986. It has been mentioned in THE MISOSYS QUARTERLY for the last three issues. What more do you want? In eight months we have sold about 1100 copies; a very dismal showing.

On the other hand, since some folks out there are putting together their own patches for date-extending LDOS, perhaps then I need do nothing for the Model I release of LDOS 5.1.4!

Files missing from Mark III Collection

Michael Kushner To MISOSYS, Inc: Hi Roy! Received the Mark III/IV Collections (with multiple copies of the LDOS Quarterly!!) and the Utility I packages okay. Thanks for your fantastically prompt service as usual.

One thing however, I can't seem to find MACH-2 programs on the Mark III disks! No ALLOC/CMD, CALC/CMD, HANDY/CMD or MAPPER/CMD. Is this normal? They're all on the Mark 4 disks though.

Fm MISOSYS: You are joking, aren't you. I'll check the master. But they are supposed to be on there. Yes, seems like HANDY, CALC, and MAPPER vanished. I'll have to see if I can squeeze them back on. As you can see, both disks have 0.OK free. It will be a challenge.

Michael Kushner To MISOSYS, Inc: In the meantime, how can a person like myself obtain these missing files? Since the disks are full, there seems little point in sending them back, and what's worse, the Post Office is going on strike, TOMORROW!

Fm MISOSYS: Ah, there's the rub. We have reorganized the disks by putting the fix files together into one file, and bundling up about 7 of the filter/ASM files into one. This gave us enough space to add the CALC, MAPPER, and HANDY files. Unfortunately, it would be better for us to refresh your disks. Since your postal service is going on strike (again???), it would do no good for you to send to me nor I to you. Let me know when the strike is over.

For you others out there needing those three files, return your two MARK III disks and I'll update them.

LDOS 5.3 and The Home Accountant (THA)

Scot Silverstein To LDOS support: Can you provide me with info on the adaptation of The Home Accountant, sold by RS for TRSDOS 1.3, under LDOS 5.3 on a model 4P at 4 MHz? The higher CPU speed and the higher file loading/saving rate would be most useful to me. I am a legitimate owner of 5.3.

jeff brenton To Scot Silverstein: Radio Shack has a sheet in the back of the manual on installing THA+ on a hard disk. This is the procedure to use when installing it under LDOS 5.x, as that is R/S's "Hard Disk Operating System". One caveat - if you DO have a hard disk, you will not have room for things like KSM or PR/FLT, as THA is incredibly wasteful of ALL computer resources, such as memory, CPU cycles and disk space.

When I reviewed it several years ago, I found it unacceptably slow, taking many times longer than MANUAL RECORD KEEPING! that, combined with the fact that it crashed the system when run under LDOS (I couldn't find enough room to have type-ahead, KSM, MiniDOS and PR/FLT in RAM and still run THA+), made me thoroughly disgusted with it. I think we re-formatted the disk.....

DEVICE status wrong for WP drives

Alex Lauf reported that the DEVICE command doesn't reflect the write protect status of drives which have a physical write protect tab affixed; however, it does correctly show WP if the drive is software protected via SYSTEM (DRIVE=d,WP).

The fix for that is SYS6G/FIX located in The Patch Corner.

LDOS 5.3 SVC table doesn't SYSGEN

Jim Howard reported that he could not SYSGEN the SVC table installed via a SYSTEM (SVC) command. The system would crash when it was re-booted. Jim was also adept at locating the bug and reported a fix. The patch is SYS7G/FIX located in The Patch Corner.

AFLAG, 2-sided SYSTEM disk, etc.

Fm Richard Politowski: I have several questions on LDOS 5.3 for which I would appreciate a direct response to concerning the implementation of the system on my Model 4D.

First, is a patch readily available which would allow the system to boot up automatically in the FAST mode without using a SYSGEN configuration file?

Second, what is the recommended method for producing a double sided disk with the system and other often accessed files clustered around the directory? I'm looking for a method similar to the "A"-flag method used with TRSDOS 6.x and LS-DOS 6.3. Is something comparable available?

Third, I am the legitimate owner of LDOS 5.1.3 purchased some time ago directly from Radio Shack. I do not have access to the 5.1.4 upgrade or documentation. Since a new manual was not made available with 5.3, is it possible to purchase the 5.1.4 documentation to keep my files complete and clarify some picky points when I am working with the LDOS 5.3 system?

Fm MISOSYS, Inc to Richard Politowski: There are three locations in SYS0/SYS which pertain to the proper setting of the FAST or SLOW speed when running LDOS 5.3 on a Model 4. One location has the value which is output to the OPREG port. The second is the value initially loaded into the time slice counter maintained by LDOS. Finally, the third location is the value loaded into the system flag, SFLAG$. You would need to change all three to boot up in FAST without using the SYSTEM (FAST,SYSGEN) command. Here's a patch to do it.

```
.SYSOFAST/FIX - Default boot to FAST
.Apply via, PATCH SYS0/SYS.SYSTEM SYSOFAST
D02,93=08:F02,93=00
D05,BC=55:F05,BC=FF
D0D,12=78:F0D,12=38
.Eop
```

Second, although we have not worked up a JCL procedure to generate an optimized disk using the "A"-flag concept, we nevertheless have something to get you started. Check out Martin Pollard's "A"-flag modifications which follow in this section.

Finally, you need to order the LDOS 5.1.3R documentation updates. This documentation set updates a Radio Shack 5.1.3R LDOS manual to 5.1.4; thus, it covers the changes between release 5.1.3 and 5.1.4 of LDOS. That set is yours for $2.50 (U.S.). Anyone not yet ordering the LDOS 5.3 upgrade kit who needs that 5.1.3R set of documentation can order it along with their 5.3 upgrade for just $2 additional.

ADDING AFLAG$ TO LDOS 5.3

Martin Pollard
21116 Erben
St. Clair Shores, MI 48081

One of the more handy features in LS-DOS is the allocation flag, or AFLAG$. This flag contains the cylinder number that LS-DOS uses to begin its search for unallocated granules; its default value is 1. This is a great thing to have when creating a system disk (or placing LS-DOS on a hard disk); since the system files should ideally be placed around the directory for minimum head movement and quicker access, changing AFLAG$ to just a few cylinders below the directory does the trick.

I decided to add AFLAG$ to LDOS 5.3, so LDOS users could enjoy the same nifty feature that LS-DOS users have enjoyed since version 6.2. This did not come easy, however. I needed an unused memory address in the resident system region (4000H-4DFFH) to test the patch, but unlike LS-DOS, the source code to LDOS has never been published. I used 4DFDH, which appeared to be an unused address (but I was not positive). As it turned out, the patch worked, but I wanted to use a real, honest-to-goodness unused address. I was stumped!

To find the answer, I wrote a letter to Roy Soltoff, who is the principal author of LDOS and is therefore the world's number-one expert on it. His reply solved the problem, as well as pointed out a potential problem that would have developed had I continued to use 4DFDH. It turns out that the 4DF0H-4DFDH area is a vector table that corresponds to the seven

modules associated with DFLAG$. 4DFDH was part of the KSM/FLT vector. Since I don't use KSM/FLT, I didn't have a problem. However, people who use KSM/FLT would have discovered a big problem: either the computer would have locked up, or LDOS would start allocating granules at a non-existent cylinder. The free address that I was searching for was 4767H; I changed my patches accordingly, and now present them to you.

INSTALLING THE PATCHES

The patches are in the form of two /FIX files, SYS0AFLG/FIX and SYS8AFLG/FIX, and are listed in The Patch Corner. The first patch, for SYS0/SYS, sets AFLAG$ to an initial value of 1, the default value under LDOS 5.3 and LS-DOS 6.3. At the DOS prompt, type:

```
PATCH SYS0/SYS.SYSTEM USING SYS0AFLG
```

The other patch, for SYS8/SYS, inserts the code necessary to fetch the value at 4767H. Fortunately, and thanks to Logical Systems president Bill Schroeder, the perfect patch area was already present, being the result of another patch!

Let me explain. In the October 1983 issue of the LSI Journal (volume 2 number 4), his article, "A Case of Mis-Allocation", presented a patch to SYS8/SYS for LDOS 5.1.3 that officially changed the method LDOS used to find unallocated granules. Previously, LDOS used a "random" method that randomly selected a starting cylinder. This created all sorts of problems, from slower access of files to files created with an excessive number of extents. The patch changed SYS8/SYS to use the "controlled" method presently used in LDOS 5.3 and LS-DOS 6.3; this method starts the granule search at cylinder 1, progressing outward to the last cylinder. TRSDOS 6.1 was released with this change made, and TRSDOS/LS-DOS 6.2 further expanded on it with the inclusion of AFLAG$.

Bill Schroeder's patch replaced a 6-byte instruction sequence with a simple LD L,01H and four NOPs. Apparently, when LDOS was reworked for version 5.3, the extra NOPs were not considered a real priority to remove, so the code stayed in. This is the area where my patch is inserted. At the DOS prompt, type:

```
PATCH SYS8/SYS.SYSTEM USING SYS8AFLG
```

Pressing RESET will effect the changes (or, more precisely, the change made to SYS0/SYS).

USING AFLAG$

To use AFLAG$, the MEMORY command is used to change address 4767H, i.e., "MEMORY (A=X'4767',B=xx)", where "xx" is the value you wish to use for the starting cylinder. During normal operations, AFLAG$ should contain 1, which is accomplished with the command "MEMORY (A=X'4767',B=1)". If the value is greater than the ending cylinder number for a particular drive, LDOS starts its search at cylinder 0 for that drive.

CONCLUSION

I would like to thank Roy Soltoff for his assistance in making AFLAG$ possible, and to thank him for the notoriety, as he informed me that he will officially reserve address 4767H for AFLAG$. That being the case, I will close my article with "official documentation" for AFLAG$:

AFLAG$ - X'4767' - Allocation Flag

> Contains the starting cylinder number to be used when searching for free space on a diskette. It is initialized to 1 on power-up. If the starting cylinder number is larger than the number of cylinders for a particular drive, searching begins at cylinder 0.

LS-DOS Model 4 Information

Accessing RS-232 from BASIC

Fm LDOS Support To Kevin Heyboer: The reason that [communications in] BASIC will work on the IBM and it won't on the TRS-80 is that the IBM BASIC was specifically tailored to support interrupt driven communications, with an internal buffer than can hold incoming characters while BASIC is off somewhere else. The TRSDOS 6 OS and Model 4 hardware do support interrupt-driven RS232 I/O, but Tandy never bothered to have the BASIC modified to properly support the features available in the OS.

This is also likely the reason that your attempts at polling the RS232 port directly have failed. If you loaded the COM/DVR, it is stealing away the characters via the interrupt long before BASIC could possibly get around to polling for the character.

To "cheat", and play around with serial input in BASIC, you can try the following. At the OS level:

```
SET *CL COM/DVR
SETCOM (as appropriate)
LINK *KI *CL
BASIC MOOSE/BAS
RESET *KI
```

where MOOSE/BAS is a basic program something like the following, and the above should probably be in a JCL. Note that you can't use INPUT or LINE INPUT in the BASIC program due to the pending JCL [you can use INKEY$].

```
10 REM Sample BASIC program for RS232 I/O
20 OPEN"O",1,"*CL"
30 PRINT #2,"This goes to the RS232 port."
40 A$=INKEY$:IF A$="" THEN GOTO 40
50 REM Now we have a character, process it.
60 PRINT A$;:GOTO 40
```

Note that you can't distinguish between real keyboard input and RS232 input. All in all, a machine language subroutine that initialized a moderate buffer, perhaps 1K or so, along with using the system RS232 wakeup vector should be useful for what you wish to do. 'C' or straight assembler is going to be the choice implementation, methinks.

RSHARD: What does it support

Fm Jeff Beck To MISOSYS, Inc: Can you give me some idea as to which controllers are supported in the RSHARD package and which ones are not? I'm trying to put together a HD set up using Ampro's SCSI host adapter and possibly a Seagate ST225N (SCSI embedded in the controller).

Is it true that the Aerocomp package won't run RSHARD? Is this because of the controller it uses?

Fm MISOSYS, Inc: RSHARD specifically supports the Radio Shack hard disk environment. That means the WD controller (1000 series). It also means specific port assignments as documented in the RS tech manual. The Aerocomp drive (the new ones) use an Adaptec controller. That's totally different from the ones RS used. The Adaptec follows the SASI/SCSI type of handshaking and uses one data port and one status/command port. I don't see us doing a driver for their drive since we don't have one to evaluate, nor would we expect to buy one - we have enough drives as it is. The Adaptec controller is the same as what Lobo used for their WIN series, but theirs was memory mapped, not port mapped. If you are putting together your own package, you will probably have to put together your own driver. Also talk to Powersoft. They have a few driver packages, also.

Fm LDOS Support To Jeff Beck: The RSHARD package supports only Radio Shack command and address compatible Western Digital drive controllers, not the WD SASI or SCSI controllers. The only drives I know of in this category are RS's and one offered long ago by Apparat.

The newer Aerocomp hard drives use an Adaptec controller, and as such requires its own special software.

Debugging bank switching

Fm Paul Bradshaw To MISOSYS, Inc: I have the XLR8er with their FIXBANK software. Recently, while trying to debug a bank-swapping program, I noticed that I could no longer use debug with another bank resident (other than 0). I've been able to step programs with other banks present before, but now, the first "I" immediately after the swap hangs the computer. I've taken the precautions of NOT using PRO-DD&T, filters, Altres, or any OTHER high$ modules during this, but the hang still happens. Could it be the FIXBANK program? Since you have the XLR8er, can you duplicate this problem? I'd like to know if it's just me, or something else. Oh yea -- on the DEBUG problem, my HD driver is in high$...

Fm MISOSYS, Inc: You invariably cannot use DEBUG to debug anything with another bank present. Where's the stack? Don't forget system device I/O automatically swaps in bank 0 - regardless of where the stack is. Practically all debugging of bank swapping stuff must be via paper and pencil and a great deal of thought (i.e. debug from afar). Afterthought. One useful way to debug with bank swapping going on is to gen a new configuration where you first set HIGH$ to 7FFFH. Then you will be ensured that nothing is in HIGHMEM. With your HD driver up there and SYS5 swapping in from disk, that's a sure crash city. Keep HIGH$ in the resident low memory at 7FFFH and you may get some success.

Fm Paul Bradshaw To LDOS Support: I'm NOT using DEBUG (EXT) -- just plain 'ole DEBUG. How are you supposed to go about debugging bank accessing and bank resident code, anyway (supposing DEBUG can't do it...)?

Fm Paul Bradshaw To MISOSYS, Inc: Well, regarding DEBUG, your HIGH$ solution sounds like the best shot. As far as my previous troubles, the Stack was in low memory (I used my own stack space), and there was NOTHING in high$ but the HD driver. Doesn't DEBUG re-enable bank zero before coming back? And then re-enable the previous bank before each "step"? It seems like doing that would solve the problem -- unless of course that requires loading in another overlay that would wipe out DEBUG (don't tell me...right?). Why would SYS5 need to keep swapping in from disk? Doesn't it just stay resident during single stepping? (I know, I know -- get The Source!)

Fm MISOSYS, Inc: DEBUG does not swap in bank 0. The interrupt processor, device I/O, and disk I/O, does bring in bank 0. This means that if something other than 0 is resident when an interrupt occurs, 0 is what's resident when DEBUG is entered from a BREAK interrupt. DEBUG doesn't come in off of disk during single stepping unless what got stepped was something that caused a DOS overlay to swap in (open, close, remove, fspec, fext, etc., ...). When I want to avoid maximum interference from the system overhead of DEBUG, I also SYSRES the SYS5 module. That then lets me step through code which expects the system's FCB to remain intact. That also eliminates alteration of the system I/O buffer. For instance, when a file is opened, the SBUF$ contains the directory record which has that file's DIREC. That's needed by certain utilities. Another thing. Try to access a PaDS member when DEBUG is on. Doesn't work. That's because SYS5 coming in off of disk over-wrote the system FCB which had the PaDS FCB info in it with DEBUG's FCB info. If you SYSRES SYS5, DEBUG doesn't come in off of disk so it doesn't effect it. SYSRES SYS5 and you can then access a PaDS member (for debugging) with DEBUG ON. Just tricks of the trade.

Preliminary reports on PC4

Fm Mark P. Fishman: Here's a brief report, now that I have been playing with Hypersoft's PC4, which is supposed to emulate a Model 4 (Z80 plus OS plus ???) on a PClone. It has a few shortcomings at present, but there's promise.

Mike Gingell, the programmer/owner of Hypersoft, will send a free update in the fall to anyone who buys Version 1.0 now. He needs feedback as to what works and what doesn't, and maybe some help -- that's my editorial, not his statement -- getting all the features in. Version 1.0 doesn't even support all the SVC's of TRSDOS/LS-DOS 6.x, so there's a lot of stuff that doesn't run yet.

Scripsit (4) works, as does PowerScript; but Allwrite doesn't clear the screen properly, and Prosoft has told me that the formatter doesn't print under PC4 yet.

Supposedly, BASIC runs, but I don't do much in that vein, so I can't say. Filters and drivers can't be loaded in version 1.0, either.

So you can see it has a way to go. I'd like to see this thing fly, but alone I don't make a market. If you'd like to take your Model 4 software over to MessDos, call Hypersoft or send EMAIL to Mike Gingell and order a copy of PC4. The price is $79.95 + $2 S&H. Definitely a work in progress...

Fm Lee C. Rice To Hypersoft: You asked for comments on the PC4, so here they are. Excuse the length of the comments. In the last issue of THE MISOSYS QUARTERLY, there was an announcement that the next issue (Vol. II#1) would contain a review [editor's note: not true, we only noted that we had seen the ad; we have not received a copy of PC4 yet for review purposes] and information on PC4. I am taking the liberty of sending a copy of these comments to MISOSYS, since some of the information may be useful to others of their readers. Hypersoft and MISOSYS are two good reasons for staying with TRSDOS and the Model 4. Your software and support are superior, and in many cases there is nothing in the MSDOS world which can touch either.

First, a warning about some of my own comments. I gather from your intro in the PC4 Manual that PC4 was intended for users who are getting rid of their Model 4 systems, but want to port some software to their MSDOS system. I am NOT in that category. I own a Model 4, Model 3, and a Zenith Z-158 (the latter is host to PC4). Since the business world is going MSDOS, I do have to do a lot of MSDOS software for departmental applications; but I find the MSDOS system quite kludgy. TRSDOS6/LDOS5 are superior programming environments by far, so I do most of my development under TRSDOS, and then port the results (after debugging) to MSDOS and/or LDOS. Most of my work is in Fortran or C, and I have suitable compilers on all three systems. My reason for mentioning this is just to make clear that my needs are probably different from those of most of your customers. Accordingly, you may decide that some of my suggestions (below) are not really important to your marketing of PC4, and you may be right...

First, some more software which runs well under PC4. Microsoft Fortran (Model 4) seems to run perfectly (F80 and L80). The MISOSYS RATFOR preprocessor also appears to function correctly, so users can have a complete Fortran4 package running under PC4. I have not tried the MISOSYS C compiler (PRO-MC) yet, but I'll get around to that in the coming weeks.

Allwrite works well also. I can't get the shift-arrow functions (beginning and end of line) to work properly under Allwrite. Also the BACKSPACE, which under MSDOS3.2 on the Zenith functions as a RUBOUT, is a simple no-delete backspace under Allwrite. I own the Allwrite Laserjet-Plus Support Package also, and that works impeccably as well.

One of the features which I miss under PC4 is the redirection and device independence which make TRSDOS/LDOS such magnificent programming environments. I hope that some future release of PC4 will implement some version of the ROUTE and LINK commands. In fact, I don't use those features that often because I have the MISOSYS Z-Shell utility, which enables redirection and piping as part of a command line "on the fly." Z-Shell is a permanent part of my TRSDOS6 programming environment. It DOES NOT WORK under the current version of PC4 - it aborts with a message that an appropriate DOS call has not been implemented. One frequent use which I make of Z-Shell is to redirect printer output (from Allwrite) to a disk file. That enables me to port the disk file to another system or printer later.

For what it is worth, this redirection (from printer to disk file) is NOT available under MSDOS either. Hopefully MISOSYS will someday produce an MSDOS version of Z-Shell. I have Superspool for MSDOS, and it functions correctly as a print spooler even when PC4 is active. Despite the fact that Superspool is allegedly one of the best print spoolers for MSDOS, it does not enable printer-to-file redirection either.

If you compile a program under MISOSYS PRO-MC under TRSDOS6 on the Model 4, and include redirection provisions in the program itself, then, after compiling (under TRSDOS6) and porting the program to the PC4 environment, standard redirection is possible. I gather that this is because the compiled C code contains the redirection runtime support already.

So my own (personal) verdict on PC4 is that it is a fine program, everything you advertise it to be, and bug-free. You indicate that there will be future updates and enhancements, so perhaps some of my wish-list items for redirection will be implemented in future releases. Incidentally, for a software emulator, PC4 is quite fast. My Z-158 runs at a

much faster clock speed that the standard IBM-PC, so perhaps this accounts for some of PC4's speed. I expected something slow and kludgy (like the CPM emulators), and was pleasantly surprised.

One note about file transfer. I own both TRSCROSS and HYPERCROSS. Your customers would be well advised to use Hypercross for file transfer. TRSCROSS is very finicky about TRSDOS disks (may have something to do with the Zenith floppy controller ...). About 40% of the time, disks prepared under TRSDOS6 are unreadable by TRSCROSS. On the other hand, I have never had a disk prepared on the Model 4 under Hypercross which MSDOS could not read.

There is one additional glitch in the implementation of PC4, and that is its use of native TRS80 Ascii files. One consequence of this is that the user cannot take an MSDOS Ascii file already on the hard drive and move it into the PC4 environment for editing, nor can files be moved conveniently between PC4 environment and MSDOS. There is a way out of this, but it is a bit kludgy also. You can use TRSCROSS to move a PC4 file to a TRS80 disk (under Image transfer), and then use TRSCROSS again to move it back from the TRS80 disk into an MSDOS subdirectory. The reverse strategy applies if you want to process an MSDOS Ascii file under PC4: move it to a TRS80 disk under Ascii transfer, and then back to MSDOS as Image.

A more efficient solution would be to provide utility MSDOS programs to do the moving, adding or deleting <LF>s in the process. I am in the process of completing two such programs under MSDOS Fortran (MS2TR.FOR and TR2MS.FOR), and should have them debugged shortly. I'd be happy to contribute them as freeware for any of your PC4 users who may want to move word-processing files between the systems (MSDOS <-> PC4) without having to move them to TRS80-formatted disks.

Again my apologies for the length of this letter (which is being done under Allwrite under PC4). I did want to cover as much ground as I could. Thanks once again for a fine piece of software. I'll look forward to purchasing future upgrades of PC4.

Do your drives appear to slowly read?

Fm Shane Dawalt To MISOSYS, Inc: While I have your attention, do your model 4s (or anyone reading this message for that matter) read your masters easily? By 'easily', I mean, does it hesitate to read or does it read the source disk as fast as it writes to the destination disk (when doing a backup). I've noticed that my system has (what seems) great difficulty reading "outside" disks, like those from MISOSYS or LSI. (This is neither here nor there but one of my drives has great difficulty formatting the higher numbered tracks around 35 and up.) Be advised, this happens even AFTER my drives (all three of them) are freshly cleaned and realigned by RS. BUT, that doesn't mean RS actually did their job correctly ... I'm fully aware of that (painfully aware of in fact).

Fm MISOSYS, Inc: Slowness may depend on the sector interleave in effect when the disk was formatted. This may be out of "sync" with your particular machine. Our TRS-80 production diskettes are duplicated on a MAX-80 using a special version of QFB which does four at once. That particular version of DUPE may not have an interleave table which is optimum for a Model 4; in fact, it may present a worse case. But then you don't run from your master disks, do you? Of course not, you make a backup at once then put the master away. Thus disks formatted on your system may be locally optimized.

JCL: When you don't have to compile...

Fm Robert G Strickland: My JCL DO file compiles every time DO xyz/JCL is run. Is this necessary? Is there a way to save the compiled program and run it as any other /CMD file? Appreciate the feedback.

Fm Shane Dawalt To Robert G Strickland: If you don't have any commands starting with "//" in your JCL, simply copy your xyz/JCL file directly to SYSTEM/JCL and execute it with "DO *". The "*" tells the system to not compile and use the info currently in SYSTEM/JCL. You may or may not know, but when your /JCL file is compiled, the system places the compiled version into SYSTEM/JCL. As long as you don't change your JCL, SYSTEM/JCL is valid.

If your JCL file contains "//" within it, then you must compile it. I'm pretty sure that once you compile a JCL with this type of commands, you may still use the "DO *". However, you cannot simply copy your /JCL into SYSTEM/JCL as you can do as noted above. I use the former case all the time when doing JCLs

for PRO-MC compilation! (I have very seldom ever used those "//" commands.)

Fm Adam Rubin To Robert G Strickland: If the only JCL macros (those "//" things) are execution macros (listed on page A-5 of the TRSDOS 6.2 manual), you can run the JCL without compiling it. Just type DO = XYZ. If you're using any of the conditional macros (p. A-15) or the "%" symbol [or "#" -editor], you'll need to compile it each time to evaluate all the conditional expressions and perform any substitutions.

"DO *" will re-execute the last JCL compiled, but with the same parameters (conditions and substitutions) as when it was compiled. If your XYZ/JCL file doesn't need to be compiled, it's much easier to use DO = XYZ than to worry about SYSTEM/JCL.

Fm T. Lee Horne To Robert G Strickland: Along with the other replies to your question, read the manual Appendix A pages 3-34 (Trsdos 6.2) from Tandy. Once a JCL is compiled ton system/jcl copy system/jcl back to original/jcl. Then execute it by DO = original/jcl. The "DO =" command bypasses compilation of the JCL. With the // Macros in the jcl it must be compiled. You can also cause a compile without execution by DO and either $ or ? instead of =, I don't remember which. If you want to compile without execution and neither $ or ? does the trick just include //EXIT before any action commands happen, you will end up with system/jcl as the compiled version of your JCL. Go into system/jcl with TED remove the //EXIT and save the program as ORIGINAL/JCL. Then any time you call it with do = original/jcl, it will execute without the delay of compiling. I have some huge jcls that I always call this way. One thing to remember. If you do = you do NOT get system/jcl. You execute ORIGINAL /JCL. If ORIGINAL/JCL calls its self by do *, you must change that line to do = original/jcl. good luck LEE

SCRIPSIT defaults: how to change them

Fm Dave Sanders: Does anyone have a patch for Model 4 Scripsit to allow it to boot with the J=N. I am not at all familiar with assembly language and find it awkward to change the format line from J=Y (the default) to J=N every time I write a letter. Any help would be appreciated.

Fm Adam Rubin To DAVE SANDERS: Try (D26,63='N':F26,63='Y'). (Use a backup, of course.) Gosh, and I thought I was the only one left who was still using plain ol' Scripsit!

@EXMEM revisited

Fm Tim Clute To MISOSYS, Inc: Is the Exmem SVC listed in the latest TMQ going to be incorporated into a future release of 6.3?

Fm MISOSYS, Inc: That question needs to be directed to LSI; however, I doubt that they will be doing any other release which would justify adding the SVC. However, if they do another release, I'm sure they won't interfere with what @EXMEM is doing or where it installs.

Fm Paul Bradshaw To MISOSYS, Inc: I would think it would be easy enough to just start distributing @EXMEM as a /CMD file that would install it. That would get the new SVC out to the most users, and allow it to get used. Anytime someone sent in a Disk to LSI for replacement (or whatever), they could just pop on a copy of it. And likewise, all disks shipped from now on could contain a copy. THEN, should they ever do any recompile, they could include it as part of the dos.

On another subject -- anyone see the latest RS Computer Center Mailer? They ACTUALLY advertised a Model 4D! I almost died of shock!

Fm MISOSYS, Inc: Yes, I saw the flyer; however, at a system bundled price of over $2000 I don't think too many people will bite. The @EXMEM SVC does install with a /CMD file which ws published in TMQ. LSI has to take whatever action from here. On the other hand, no one (repeat NO ONE) has taken me up on my offer to develop a BASIC interface as input to TMQ. No one has supplied me with ANYTHING which will utilize the @EXMEM svc. All I have heard is bellyaching as to when LSI will put it into the DOS, when I will get it onto the forum's DL, etc. When is someone going to demonstrate some more uses of @EXMEM for a QUARTERLY article? When there is a demonstrated NEED for @EXMEM added to 6.2 by LSI, they would be more akin to do it. I developed the idea of the SVC and implemented it. Let's stop bellyaching and USE it. Then we'll see where LSI can take it. How about it?

Fm H. Brothers To MISOSYS, Inc: I haven't been one of the complainers, and I have been playing with @EXMEM in Basic, C and assembler. One question I keep asking myself, however, and which I haven't yet had the time to test, is whether @EXMEM is more efficient than a good RAMdisk and, if so, by how much. What I need is a quick way of sorting more records than will fit into normal memory. But both @EXMEM and a RAMdisk require that the records be brought into memory, swapped, and then returned to a back bank. What I guess I really need is another utility that can go into a bank and rearrange things there directly, without the necessity of an extra set of transfers through low memory to regular memory then back to low memory and back to a bank. Any suggestions?

Fm MISOSYS, Inc: Actually, a RAMdisk driver should be rewritten to utilize @EXMEM. If you think about it, memDISK has a lot of what I put into @EXMEM. By using @EXMEM, memDISK could be cut probably in half of what it is currently using out of low memory. In fact, memDISK could then be located in high memory. @EXMEM is best for conserving low memory for things which must be low. Shuffling data through memory is not necessarily inefficient if you make use of full page transfers. Trying to do byte I/O is going to be slower because of all the overhead needed to just set up the transfer. You might experiment in using pages.

Now another method available to the assembly language programmer (or very skilled C programmer), would be to write the re-arranging code to be executable in an alternate bank. Stuff the code, stuff the data, then go to it. That would cut down on a lot of data shuffling through memory. That's the utility of the optional transfer address passed in HL in the @BANK service call. That's tough to write, though. I used that function for the SPOOLer to enable it to run in alternate memory with a small front end interface in bank-0. That technique could actually be employed to write very large programs, although I wouldn't recommend it. You know, if you examine the Extended Memory Manager (EMM) developed for the MS-DOS environment, you can see good similarities in @BANK.

Fm H. Brothers To MISOSYS, Inc: I probably will do some more experiments after this month's deadline crunch is past. From about the 5th to the 20th of each month, I have precious little time to experiment or program -- then things usually lighten up a bit until the next month rolls around, and I get to do a little programming for fun.

A ramdisk written on top of @EXMEM makes a lot of sense as a way of avoiding low-memory congestion. Having @EXMEM, a ramdisk, and Pro-WAM all sitting in low memory seems redundant, to say the least. Any chance of having a high-memory form of Pro-Wam as an option in the next update? (Well, at least I can dream about it, can't I?)

Fm MISOSYS, Inc: PRO-WAM only uses about 25-35 bytes of low. True, it could operate entirely through a facility such as @EXMEM. I just don't want to encumber "green" PROWAM users with something else to install.

Fm H. Brothers To MISOSYS, Inc: Two questions about @EXMEM. The first is about the coding of the program itself. Wouldn't it be more efficient, in terms of low memory required, to call MOVSTAK at the entry to @EXMEM instead of once inside each function? I may be missing a subtlety here in the way @EXMEM operates, which is why I'm asking the question.

Second, you mention in the winter TMQ that you may write a version of MEMDISK using @EXMEM for a future issue of the quarterly. Is that still in your plans -- if so, I'd like to mention it in my article. If not, I'll simply keep quiet on that subject.

Fm MISOSYS, Inc: Since you just mentioned it, I did a quick peek back at the code. I don't see any particular reason why you couldn't move the CALL MOVSTAK to the EXMEM@ entry. For that matter, it wouldn't even have to be a CALL, just swing the code in and revamp it to get rid of the save/restore of the caller's return address. The only thing I see off the bat is that interrupts would be off for a little bit longer and the stack would be switched around even on a bad function call; however, I don't see that as derogatory. I believe that on the other point, I suggested that it would be very easy to rewrite memDISK to use @EXMEM - I didn't (at least I don't think I said) say that I would do it. Maybe in today's Ollie style, I should say that I don't recollect saying... In any event, don't even suggest that I'll be doing that - I really don't have that kind of time.

Fm H. Brothers To MISOSYS, Inc: I MIGHT put together a RAM disk based on @EXMEM, then. For my personal use, if nothing else, since right now I only have room in low memory for any two of: RAM disk, @EXMEM, & Pro-WAM. Things get a

little tight down there, and rebooting to select a different startup JCL for each job isn't a whole lot of fun (I know -- there are easier ways, but ...).

Thanks for the clarification on the MOVSTAK call. If I make that change, I might just generate enough room to be able to get everything to live together in harmony. I was also thinking about putting some of the @EXMEM code low and the rest high -- again, just for my own use.

Fm MISOSYS, Inc: It really wouldn't be tough to do a memDISK based on @EXMEM. I just don't (didn't) have the time. Just a last minute thought about moving MOVSTAK: double check the low-memory stack space. I believe I calculated it based on exact usage.

LSI Patches: Can you port them?

Fm Ken Peck To MISOSYS, Inc: What is the copyright status of the LS-DOS 6.3 patches on the Disk Notes? Is it permissible to upload them on a BBS? I really am trying to get folks to subscribe to TMQ, but some folks around here are asking for the patches.

Fm MISOSYS, Inc: The LSI patches that were published in DISK NOTES are on our forum. Virgil put them up some time ago. Also, the MISOSYS patches are on the forum as FIXES5/FIX, FIXES6/FIX, and FIXES7/FIX (or could be /TXT).

Getting RSHARD6/DCT in to high memory

Fm Dick Newman To MISOSYS, Inc: I have been unable to get RSHARD6/DCT into high memory. I am using the XL8er board and FIXBANK, FIXALL, and RAMDISK are in low memory. TRSHD6/DCT is in high memory and works fine. When I attempt to install RSHARD6/DCT in high memory the system locks up and I have to reboot. Is low memory "core" an obscure word or an obscure place?

Fm MISOSYS, Inc: RSH62/FIX in TMQ I.iv fixed up that problem. I believe at this stage of the game, "core" is an obscure word in referencing memory.

Fm Bill Schaper To MISOSYS, Inc: Thanks for the RSH62/FIX patch, Roy. It works fine with all filters now in low memory and RSHARD6/DCT in high. As I was telling Jim Gaffney, I sysgened the whole mess and it works fine. Thanks again for the support and you can count on my support from this end.

Why you should avoid extraneous date patches

Fm Alan H. Pesetsky: I note that some folks in Canada (Michael Jacobs Consulting) have developed a series of "Shareware" patches for TRSDOS 6.2 as an alternative to LS-DOS 6.3. (allowing dating into the year 2000 and beyond) and deposited them in the DL's of the TRS Professional Forum. Has any one tried them? Any comment of the feasibility or advisability of using such or the possible reasons for developing it?

Fm LDOS Support To Alan H. Pesetsky: I see several problems with this, most apply also to the set available for LDOS 5.1.4:

(1) The patches do absolutely nothing to solve the problem, they merely treat some of the symptoms. Namely, they change the base offset for the start of the eight year range available in the directory. You might as well just lie to your computer as to what year it is. Just as effective, and you don't have to kick in for the shareware registration. What about existing disks?

(2) The distribution of patches like this as shareware is questionable. Compare the shareware contribution against the value received with the new OS versions. Added utilities, features, augmented BASIC in the case of LS-DOS, etc. are really bargain priced as it is.

(3) They might give many users a false sense of security. They get the patches, then hold off putting them in 'til '88. When they do, they find out the drawbacks and reconsider. Problem is that because the patches cut into sales of the new OS versions, they aren't available anymore!

Fm Robert G Strickland: I heard that there are some patches to TRSDOS 6.2.1 which allows using the date stamp beyond the end of this year. Would someone please direct me to the proper files, and is there any hidden drawback to this approach over buying 6.3?

Fm Paul Bradshaw To Robert G Strickland: Yes there is a draw-back. You loose the ability to handle early files (i.e. the patch just changes the "base" year from '80 to something

higher). You also lose time stamping, date/time displays, and many other misc. enhancements and fixes. Go the "correct" route and get LS-DOS 6.3. You won't regret it, and your show of support to Model 4 vendors will keep them in business just that much longer, and will keep Mod 4 software on the shelves just a while longer.

Getting around disk errors

Fm Shane Dawalt: No, this isn't a bug report ... settle back down in your chair. Actually, I wanted to know why the COPY command in LS-DOS 6.3 (and in earlier versions) doesn't copy parts of a file to the destination file when an error occurs? I have a disk with some VERY useful stuff (which would take forever to retype) that has decided to go into the trash can. I can copy most of the files off of it except one. It is riddled with parity errors, but at times it can be read easily by the DOS. Anyway, I tried to copy it over to another disk and all I get on the destination disk is the appropriated directory slot along with 0 values for records and EOF. It shows 6K of space used but there is no data in the file. This happens when a parity error occurs. I don't understand why COPY doesn't just copy each sector as it comes to it. When an error occurred, all the data up to that point would be saved.

Fm MISOSYS, Inc: When an error occurs, most likely the destination file doesn't get @CLOSEd; thus, the directory is never updated to indicate any size. Your best bet is to use MONITOR which was part of the PRO-ESP package now bundled into MARK 4 Collection. Monitor slips in between the disk driver and the DOS. When a disk error occurs, you get something akin to the BIOS trap in CP/M and subsequently MS-DOS - the "retry, abort, ignore, continue". Thus if the disk is prone to parity errors, you can use MONITOR's "retry" to essentially give you infinite repeat - until you wear your finger out hitting the "R" key. You could also use the "ignore" function so that the program doing the I/O would ignore that disk error and continue. Monitor has saved quite a few folks.

Fm Ray Pelzer: When I used to get that on customer's PROFILE files on Mod IIs a few years ago at RatShack, I used to salvage 'em by the following program (assuming bad file in drive 0 and good file in drive 1):

```
10 OPEN "R",1,"TESTFILE/DAT:0"
20 OPEN "R",2,"TESTFILE/DAT:1"
30 FIELD 1, 128 AS A$,128 AS B$
40 FIELD 2, 128 AS C$,128 AS D$
50 ON ERROR GOTO 1000
60 FOR X=1 TO LOF(1)
70 PRINT X
80 GET 1,X
90 LSET C$=A$:LSET D$=B$
100 PUT 2,X
110 NEXT X
120 CLOSE:END
1000 LSET C$=" ": LSET D$=" "
1010 RESUME 100
```

This will fix it (you may need to CLEAR string space). Actually, the term "good" file above is incorrect; Drive 1 would END UP holding the salvaged "remains" of the bad file.

Faster operation with SYSRES

Fm Theodore Masterton: I would like to know which sys files should/could be rammed in order to speed up disk access. If there are trade-offs (and I would be surprised if there were not), I would like to know which sys's do what re: disk access. I do have sys file functional descriptions from the old LS quarterly and the documentation but frankly references to things like "vectors" are over my head.

Thank you so much. Oh yes, I would like to have these information as it pertains to LS-DOS 6.3 & LDOS 5.3. -

Fm LDOS Support To Theodore Masterton: Some of this is going to be very condensed, ask for more details on anything you want. Anyway: This information applies to LDOS 5.1, LDOS 5.3, TRSDOS 6.x and LS-DOS 6 with one exception: In the Model 4 mode systems, SYS8 is a library file and is used far less frequently than SYS8 in LDOS, which is used constantly for file allocation.

SYS0 - Used only for booting. If you are tight on space, you don't even have to have this on a working (but not bootable) system disk. Of no use to run on a ramdisk, can't be sysresed. Should be moved to a ramdisk and then removed in the 6 systems, to move system defaults.

SYS1 - Command interpreter, does the DOS Ready thing. Needed always, should be ramdisked, low priority on sysresing 'cause it is not called as often as some others.

SYS2, 3, 8 (5 Doses only) 10 - These are the file handling gizmos. Needed always, should be ramdisked, high priority on sysresing for fast disk I/O performance.

SYS4 - Error messages. Delete this for disk space, but then everything is a "SYS Error". Since you don't get errors (wishful thinking), very low priority for sysresing except for some "tricks" (BACKUP).

SYS5, 9 - Debug and Debug (E). Dump 'em if you don't do assembler programming.

SYS11 - JCL. Needed for execution of JCL, should be ramdisked. Very low priority for SYSRES, JCL execution has to go to disk anyway.

The LIBs (SYS6, 7 and 8 (6 Doses only) can't be sysresed, but should be ramdisked if possible so you have the LIB commands available. A LIB will show you A and B for SYS6 and SYS7 (and C/SYS8 for 6 Doses), dump what you can live without if you need the space.

Using 3.5 inch drives with LS-DOS 6.3

Fm Duane Saylor: Now LSI has said that LS-DOS 6.3 doesn't support 3.5 inch drives (in MISOSYS Quarterly). I have been running an NEC FD1035 under LDOS 5.3 and TRSDOS 6.2 for 9 months!

Fm Duane Saylor To LSI: I have been using LDOS since it first became available. I have been using this forum since it began. I have always had the highest regard for answers provided by JJKD, Tim Mann and many other users. LSI has provided an excellent operating system and excellent support. When they licensed Tandy to provide TRSDOS 6 I was elated but after many months waiting for Tandy to provide new versions I wished that I could get the updates directly from LSI. When LSI chose to provide a version with correct dating to 1999 I was pleased. I delayed in ordering because I use the Alpha Technology Ramdisk software that needs @BANK patches and I was not sure when or if @BANK patches would be available.

Another reason for my concern in using LS-DOS 6.3 is the statement in the current MISOSYS quarterly in the LSI LS-DOS column. "Because of several hardware differences, it would require a completely new driver be written to support these (3.5 inch) drives." It may come as a shock to you that I have been using an NEC FD1035 drive in my Model 4 under both LDOS 5.3 and TRSDOS 6.2 since October 1986. Installation required mounting a 34 pin socket type connector on the disk drive cable and making a power adapter. After installation a "FORMAT :1 (SIDES=2,CYL=80)" was all I had to do to format a 3.5 inch disk. In fact I can do mirror image backups of 5.25 inch 80 track disks to 3.5 inch disks. I have had zero problems with the NEC drive in the last 9 months. Could you tell me why the disk driver in 6.3 was changed so it does not support 80 track double sided drives?

Fm MISOSYS, Inc: The @BANK patches for the Alpha Board and 6.3 have been on the forum for about 8 months. They were provided before 6.3 was released. They also appeared in an earlier TMQ.

Fm Doug Mayfield To LSI: According to the paragraph in TMQ I.iv, you say something to the effect that LSI (LS-DOS) doesn't support 3.5 inch drives (720K). There have been some messages [on the MISOSYS forum (CIS)] concerning this. My question concerns this statement. I had been planning on getting a 3.5 inch drive in the future (720K) and was wondering if this was a misprint, and what you were actually thinking of was the 1.2 meg drives (3.5"). If that is the case please let me know so this will be cleared up for all concerned. I didn't think there would be any problem with the 720K drives, as I assumed they would act just like any other 80 track, double sided drive, but I don't really know. Thanks for your advice and support of the Model 4.

Fm LDOS Support To Duane Saylor: I was a bit puzzled by that statement myself. I can think of only two possibilities. First, there are some differences in newer three and a half inch drives, namely change-line support on pin 34. This is a pretty easy hardware fix, however.

Second, on some of the original three and a half inch drives, @CKDRV needed to be modified to allow for a possible increase in the delay between the time the motor and select get turned on and the point at which the media actually starts rotating (index pulses). I developed some (successful) patches for the TRSDOS 6.2 release to cover this, and these are in the DLs here. LSI may not wish to officially support this kind of patch, but it shouldn't be hard to develop.

Fm LSI To Doug Mayfield: That LSI "doesn't support" something doesn't imply that it isn't available or that it doesn't exist or that it

is impossible. Witness the many non-RS hard disks that work just fine without our support.

Fm Duane Saylor To LSI: The announcement that LSI is not going to write a special driver for 3.5 inch drives also implied that one was needed. At least for the NEC FD1035 under TRSDOS 6.2, a special driver is not needed.

Fm Kevin R. Parris To Duane Saylor: Please tell me more about your experience (and configuration) with the three-and-one-half-inch disk drive and the Model 4 (or 4/p?). Things like how is it connected to the system, where did you get it, and what did you have to do to the computer to make it talk to the drive. The prospect of using 3.5 instead of the old traditional 5.25 sounds pretty good to me. Waiting eagerly to hear from you!

Fm Duane Saylor To Kevin R. Parris: To mount the drive in my Model 4, I took an old discarded 5.25 inch drive and using a bandsaw cut the middle out of it and made one bracket to mount the 3.5 inch drive where I cut the middle out of the old drive. It would probably be more practical to get a drive like the Toshiba 3.5 inch drive which already has the 5.25 inch mount and electrical interface. With the Toshiba drive you simply mount it and connect it with the existing connectors. With my drive, an NEC FD1035, I had to purchase a 34 pin socket connector which I mounted on the existing cable. I also made a power adapter cable because the 3.5 inch drive uses a different power connector. Once the drive is mounted the command:

```
FORMAT :1 (SIDES=2,CYL=80)
```

will format the drive. If you then do a DEVICE command followed by a SYSGEN, your system will boot the next time knowing that that drive is an 80 track double sided drive with 720k of storage.

Fm Kevin R. Parris To Duane Saylor: Let me see if I understand you correctly: If I got a Toshiba drive, I just unplug the existing floppy drive and hook up the new 3.5 drive in the same space? That sounds almost too good to be true! By the way, I have a 4/P- do these new drives come sized to fit in the space of the existing drives?? Thanks much for the information!

Fm Duane Saylor To Kevin R. Parris: Yes, the Toshiba drive that you can find advertised in Computer Shopper, comes with a mounting kit that will directly replace a 5.25 inch half height drive. You can mount it with existing holes, plug in the cables, and format it for 80 tracks and two sides.

Fm David Hall To Duane Saylor: I saw that you had [answers] about using 3.5" drives under LS-DOS 6.3. Just thought I'd let you know that I too have switched over to 3.5" drives (Epson and the ones from TigerTronics ... don't remember who makes them). I am currently using them under MM CP/M (well ZRDOS+) LDOS 5.3 and LS-DOS 6.3 ... all with no problems. As with you, I needed only tell the DOSes that the drives were 80 track double sided drives.

Fm Kevin R. Parris To Duane Saylor: Thanks Much for the details! This new information from you sets my outlook on the future into a much brighter light. I had been thinking about getting some double-sided (and maybe eighty track) floppy drives, but now I figure "Why bother with that, when I can have the 'newer technology' of the 3.5 drives?". Since you say it works with no software changes, and fits in the same hole, the 3.5 comes out the clear winner. Can't get rid of 5.25 completely, though, with interchangeability (and my existing collection!) to consider. One more thing; have you ever tried putting the 3.5 in as physical drive zero to BOOT with?

Fm Duane Saylor To David Hall: What concerned me was the statement from Virgil of LSI in the MISOSYS quarterly that LSI would not be writing a driver for 3.5 inch drives. Glad to know that LS-DOS 6.3 still works with the 3.5 inch drive.

Fm David Hall To Duane Saylor: I took Virgil's statement to mean that LSI would not be writing drivers for the newer higher density 3.5" drives.

Fm Duane SaylorTo Kevin R. Parris: No, I have not booted from a 3.5 inch drive but see no reason why it should not work. I have switched to the 3.5 inch drive with a SYSTEM (SYSTEM=d) command and that works fine.

Beware of wild pointers

Fm Shane Dawalt: I just got done attempting something which I thought would work, but it didn't (by any stretch of the imagination). I issued the following commands to TRSDOS 6.3, level I:

```
ROUTE *TT OUT/DAT:3
LINK *DO *TT
```

FYI, drive 3 is a MemDISK drive. After issuing these commands, I could not type anything on the keyboard and have it displayed on the screen. DOS appended a CR to each KEY being pressed and tried to search for it like a program. It wouldn't display LS-DOS Ready, but would always display a 'P' for it's prompt. I couldn't reset it, so I had to reboot the computer. I issued the same commands, this time using drive 2 (a REAL disk drive). Everything worked fine. I can't see why MemDISK is any different from any other disk drive ... unless it's driver is screwing something up when it gains access to the higher banks in memory. Also FYI, MemDISK was set up for both banks (selection D).

Anyway, the question is, is it legal to use a MemDISK defined disk for routing such as this?

Fm LDOS Support To Shane Dawalt: Seems to me that it should be legal. Certainly, if it works with a "real" drive, it should work with a memDISK drive. Sounds like an interaction between the banker code and the byte I/O routines. Have you tried the same setup with the 6.2 release? If the same thing happens there, don't be optimistic about it getting changed. What other drivers and filters do you have on the various devices and where are they in memory (high vs. low)?

Fm Shane Dawalt To LDOS Support: I had Forms/FLT filtering the *DO (I was using the XLATEr to change ESC codes to '.'). Anyway, I attempted to duplicate the error today and I can't. I was tempted to write it off as a stupid error. But I remember I had to reset the computer, which would have erased any problems in low core. This didn't effect the error, it continued whenever *DO was linked with *TT which was routed to a datafile on the MemDISK.

Nothing else was in memory (I don't like to bog the system down too much, especially when running PRO-MC ... which I was doing that night when this problem showed up.). I thought about my C program screwing up low core, but I just reran it with the system setup of that night, and everything worked just fine. I'll keep looking for that bug. [As Joe says, "Don't look too hard."] I WAS able to duplicate that error many times last night. (Perhaps I should wait till dark??)

Fm LDOS Support To Shane Dawalt: Sounds more like a wild pointer blasting something in low core. Whatever the pointer happened to be initialized to, perhaps based on the phase of the moon (watch out in 28 days or so).

Fm John Garner To LDOS Support: I think you hit the nail on the head there. I remember a program I wrote in LC on my MAX-80 under LDOS 5.1.4. Crashed EVERY TIME I ran it. I finally took it to a friend's house and ran it on his Model III. Surprise! It ran, but printed garbage from ROM instead of what it was supposed to. Turned out it was trashing memory at location X'000D' (yep, carriage return). On the Model III nothing happened since it was ROM, but on the MAX it is RAM and part of the @KBD code. CRASH! Even now at work, I have some program (one of about 45, but I don't know which one) written in TRS-80 Pascal which trashes location 0. Issuing the BOOT command crashes the system by switching the 4P boot ROM in with interrupts on. I found several programs which did this, but I obviously missed one. (The problem was the author passed a parameter to a routine which expected it to be VAR, but he forgot to declare it VAR in the calling routine. It is only used in case of a disk error, which never happens, so the problem never showed up.

LINKing *DO to a disk file

Fm Jim Beard To MISOSYS, Inc: Roy, [we have] a RATFOR customer who has run into the JCL problem. Basically, the problem is that the JCL will crash if you link *DO with *FS where *FS has been set or routed to a filespec, when the file does not pre-exist. The CREATE, REMOVE, etc. are attempts to get around this problem. If we can get JCL to work with LINK *DO, that is the real solution. The problem happens with 6.2, although I admit I haven't tried it with 6.3 or 5.x.

Fm MISOSYS, Inc: That will nearly always crash the JCL execution. Here's why:

When a command is completed while JCL is active, the @EXIT is handled by SYS11, not SYS1. The handler in SYS11 will check to see if the JCL file is at its EOF. If it is, it unhooks and passes control back to @EXIT which then goes to SYS1. If the JCL file is not at its EOF, it requests a @KEYIN of 79 characters. The line input handler, knowing that JCL is active, passes its single key requests to another entry in SYS11/SYS, not to @KEY. The

key handler in SYS11 will issue an @RREAD of the current JCL sector only for the first character of the input line; nothing else is supposed to get in the way of the JCL file buffer during the KEYIN of a single line. SYS11 uses the system buffer for its file buffering needs.

Got that so far? Well take a look at what KEYIN does for each character fetched. It @DSPs it! Now you have *DO linked to a device which is routed to a disk file. If the file is not pre-created with SPACE sufficient to hold the JCL display job stream, the file access routines will have to perform I/O to the directory to dynamically increase the "video" file size. Guess what? That uses the system buffer, too! That crunches the contents of the system buffer which SYS11 thought contained the JCL file's current sector.

Considering the scenario of:

```
ROUTE *TT RAT/DAT
LINK *DO *TT
DATE
```

if the file, RAT/DAT, does not exist, the ROUTE command @INITs it. This creates a NULL length file. @EXIT is requested after the LINK command completes. SYS11 passes a @KEYIN request for a new command line. @KEYIN passes a key request to SYS11. SYS11 says, aha, this is a new input line so let me re-read the current JCL sector into the system buffer. It then @GETs the next character and passes it back to @KEYIN. Well @KEYIN displays that character via @DSP which goes through the *DO device. The device handler says, hey, this is LINKed to *TT, so it passes the @PUT of the "D" character through the file access routines, since *TT is a FILE! Now the file access routines say, hey, this file has no space; let me get some. Well first I have to read the GAT to find a free granule. Then I have to update the directory record of RAT/DAT to show it is using that granule. That I/O uses the system buffer, too. Now the system buffer no longer contains the current JCL sector. The space is obtained, the "D" is @PUT to the file, and the device handler returns to @KEYIN. Since the character was not a CR, KEYIN goes after another key from SYS11. SYS11 says this is not the first character so I don't have to do a @RREAD. Could of fooled me! The JCL buffer (alias the SBUFF$) is kaput. Garbage is read, not the rest of the command line, "ATE"; it's not actually garbage, since it is a directory sector - take a good look. Well KEYIN completes, the command is hokey, and JCL ABORTS (after displaying the next 78 or so characters fetched from that directory sector which screws up the video screen).

Solution? Easy, got a spare 256 bytes to play with. Re-assign the new 256-byte page to be the JCL file buffer so there's no interference with directory access. Or keep that *DO-linked file CREATEd. Remember the requirement to CREATE the ZSHELL pipe files? Same reason, almost. Also true for Job Log files!

The LSI Column

Once again I must remind all readers that the material presented in this column is from LSI and are not opinions, positions or policies of MISOSYS or Roy Soltoff. Any contact with the author of this material should be directed to LSI management, as they assume full responsibility for the content. PO Box 55235, Grand Junction, CO. 81505 Phone: (303) 243-7070.

Things have been very busy at LSI lately, with thousands of LS-DOS 6.3 product streaming out the door. This of course is in no small part related to the fact that Tandy has sent an update notification to all Model-4 owners who are registered with 6.2.1.

Tandy and LSI have reached agreement on several points. From now on both LSI and TANDY will be supporting LS-DOS 6.3 as the current operating system for the Model-4. To assist Tandy in their ongoing support effort and to assure users of long term support LSI has provided Tandy with complete source code to LS-DOS 6.3 as well as system generation tools. All Model-4 LS-DOS product will still be produced and distributed by LSI. Tandy will however create and produce LS-DOS product for use on French and German keyboard versions of the Model-4, and distribute these versions in Europe.

The current production level of LS-DOS is - 6.3.0/Level L - with file dates of 07/01/87. There are no known problems existing in the system at this time which have not been corrected. If you are not current check the patches in past issues of TMQ.

In the August issue of 80-Micro an article appeared by Hardin Brothers which discussed the piracy protection scheme that is employed by LSI on LS-DOS 6.3. As some of the informa-

tion that Hardin published may have been misunderstood by some, there has been some unneeded concern with some of our users. Since that article we have been deluged with questions from users. Let's look at some of the facts surrounding the issue.

First, our license agreement under which we provide LS-DOS is very clear, each purchase of LS-DOS grants the right for the use of the system on ONE computer. The word "one" is used with it's standard English definition intended, that of a SINGULAR unit. If LSI would have meant that the license be granted to a USER the license agreement would have said ONE USER instead of ONE COMPUTER. Apparently general "literacy" is a greater problem for some of our users than computer literacy.

We have a very clear license and a very clear warning posted on every copy of our product (page 16). This page has accompanied EVERY copy of LS-DOS. We have no intention of condoning ANY violation of our license, no matter what rationalization the user may contrive. LSI is tired of piracy and the constant violation of the rights and direct theft by users of the property of others. We are sick of people calling with "Well I have two machines and I only use one at a time so I should only have to have one copy of the DOS"..... Ya right, and Tandy just gave you that second machine, for free, to make things more convenient for you. Maybe you just walked into the store, picked it up, and walked out the door; after all you were only going to use one machine at a time so you only have to pay for one, right? The point is that any rationale or "excuse" for violating our license is just plain stupid. If you want to violate the license, go ahead. We have no practical way to stop you. But, be aware of the warning in the documentation and do not attempt to hold LSI responsible for the results.

We have even had some idiots out there threaten legal action for one hair brained reason or another. I would very much invite any such confrontation (the free press coverage would be a blessing) and LSI could always use the money we will collect from the countersuit. Believe it or not, authors of software do have some rights and we are WELL within them.

We are getting a vast amount of questions regarding any number of hypothetical situations and real situations where people are asking to violate, or are in fact violating, our license and want assurances that it will be all right with LSI. Well it's not ALL RIGHT with LSI under any circumstances, period. What has become abundantly clear by these inquires is that the problem is much deeper and more pervasive then LSI could have imagined.

We have made this same statement on a repeated basis and the situation has not changed....

> "IF YOU DO NOT PERFORM UNAUTHORIZED PATCHES TO THE LS-DOS 6.3 SYSTEM OR TAMPER WITH THE SERIAL NUMBER OR ID NUMBER ON YOUR SYSTEM and USE YOUR SYSTEM ON ONLY ONE COMPUTER, YOU WILL NEVER ENCOUNTER THE LSI PROTECTION SCHEME!!!"....

That's it... a clear simple statement that is the "official" LSI statement on the subject. Don't ignore it... but on the other hand don't read into it what is not there.

The software industry is in serious jeopardy. With this very high percentage of users who are willing to steal from our company with such little concern, I have decided that NO future LSI software product shall be released without a hardware key. The product will then be able to be used in only the machine to which the KEY is connected. I really feel sorry for the legitimate users out there, those who are not thieves. But I have to lock my car, my office and my house to protect my property from the thieves, not the honest people. So in the future, I will lock my software. I hope this satisfies those of you who refuse to honor a simple and fair license agreement. I will probably sell a lot less copies of my future products but there are many pluses. The prices will be much higher, support will clearly be to legitimate owners and we will be able to provide better service to fewer people.

I do not believe LSI will be alone (there are already hundreds of "KEY" locked products) in making this move. LSI's future viability requires this to be done.

One last point about our protection scheme on LS-DOS. It will never damage a user's data under any circumstances, as written. If our system is tampered with it could do "who knows what". LSI can only be responsible for what we wrote. So if someone says they have a "patch" to turn off the protection.... don't use it, you may be toying with disaster. Be assured there is NO SIMPLE PATCH that will overcome our protection scheme. Also I should point out

that not one legitimate user has had any problem whatsoever with our protection system, this is as intended. We have taken great pains to protect our licensed users.... but violators and pirates be damned.

On to other things... As of this writing there are no known problems with LS-DOS. So if you are running with a 6.3 level "L" (with file dates of 07/01/87) you need not be concerned about updating your system. If your system is older you should send it in to our update department with the $5.00 update fee and we will recut your master. You can also apply the patches contained in past issues of TMQ, as they apply to your version. The choice is yours.

I personally have been very pleased with the small amount and nature of the bugs that have been uncovered in LS-DOS. There have been none that I would call real serious, most have been just annoying at worst. If you are familiar with large software works as complicated as an operating system you would greatly appreciate this observation. I am also greatly pleased with the upward compatibility that we have maintained. Other then a handful of very poorly written programs all software that ran on 6.2 seems to run perfectly on 6.3. Let us know if you find anything that seems to have a problem. Before contacting us with a compatibility problem don't forget to go back to 6.2 and check that the problem is not there also. We find almost all complaints about something not working on 6.3 that "worked fine" under 6.2 have the exact same problem when the same sequences are gone through on 6.2 as it has in 6.3. Definitely a bug in the program, not the operating systems.

For those of you who have read the article in 80-MICRO regarding the last 10 years of computing as it related to TANDY, please consider that you have read a mostly fictional account of those times. I was personally involved with TANDY as a vendor during that entire period and can tell you that that article was a "pure garbage" account of what happened. I won't go into the true history in this issue, but I probably will write it and publish it. When I do it will be factual, accurate and interesting. As delivered in 80-MICRO it was a distorted collage of meaningless material.

For this and other reasons of content LSI has decided to stop advertising in 80-MICRO. We may advertise in that publication again but that does seem unlikely.

LSI as well as MISOSYS have come to realize that the MS-DOS world seems to have become the only area that has a future. We have both treaded softly as we test the waters of this marketplace. I even tried a "SHAREWARE" release of my MS-DOS FED product. I would have starved to death had that effort been more then just a test of that method of marketing in the MS-DOS world; I did NOT get much back at all.

LSI has no future projects being developed for the Model-4, but we will continue to provide LS-DOS 6.3 and support this product as needed, or as long as the money holds out whichever comes first.

We are developing vertical products for the MS-DOS market place and are closely watching the UNIX market also. Our future work is being done in the "C" programming language for this reason. Just recently I came across some information on the new operating environment to become available from Digital Research for the 80386. This looks very interesting and should be watched by those of you who are not fans of the design of MS-DOS (and who is?).

While we are into the MS-DOS world I would like to give my opinion about the PC compatibles out there. Tandy has a fist full but I feel that you can do much better by buying some of the others. "CLONE" computers come from the same people that own and operate AEROCOMP. We run several of their TURBO/CLONES and have had NO problems. Not only is the company a known name with a great mail order reputation and a good service department but their prices are rock bottom (for first class machines). So if you are considering a PC XT type compatible, give John Long at CLONE COMPUTERS a call, tell him Billy from LSI sent you.

When it comes to AT compatible and up look to AST RESEARCH and PCs LIMITED (now DEL COMPUTING). Roy uses the AST and we use the PCs LIMITED. Both of us have been satisfied with the quality, pricing and performance of the machines.

We do not at this time run any TANDY compatibles, but we will be taking on a couple of the new Model-1400s. This seems to be the neatest little machine we've seen.

For those who program in BASIC on the PC I have provided Roy with a powerful input subroutine for use in your programs. ENJOY.

MS-DOS Information

Brief note on Microsoft QuickBasic 3.0

Fm Marc Nowell To LDOS Support: At Long Last! The QuickBasic 3.0 arrived today! It does work with RAM-resident software, and does have an overstrike mode. Why ANYONE would assign the insert/overstrike toggle to <Ctrl><O> is beyond me, though... The new CodeView style debugger is nice, especially the watch variable and breakpoint parts. The upgrade contains four diskettes: Two for regular setups and two for computers with math coprocessors.

Fm MISOSYS, Inc: We also recently acquired QuickBasic 3.0 and expect to be revising our invoicing system to be compiled under QB. Perhaps we may have some preliminary feedback by the next issue of TMQ.

Numeric coprocessor bug in PC-DOS 3.2

Fm Jim Beard: I found an MS-DOS 3.2 bug in the Turbo C (Borland) manual. When you have a floating point error like divide by zero or overflow, the interrupt processor allocates stack space to the error routine, which is not reused or reclaimed, even when you drop back to DOS. If you have an 8087 and run a program with overflow errors several times, the DOS will eventually crash. I do a LOT of 8087 number crunching, and this has happened to me several times. I was thinking of taking the box in for diagnosis.

Fm MISOSYS, Inc: The Microsoft QuickBasic 3.0 manual also notes this bug, although they state that it is in "IBM PC-DOS Version 3.20"! What"s the matter? Is it not in MS-DOS 3.2? They do provide a PATCH to the DOS. Here is what they have to say about it.

> "IBM(R) PC-DOS 3.20 Patch for Microsoft(R) QuickBASIC Compiler, Math Coprocessor Version (C) Copyright Microsoft Corporation, 1987
>
> There is a bug in IBM(R) PC-DOS 3.20 that prevents floating-point errors, such as division by zero, from being correctly handled in QB87 programs. This patch to IBM PC-DOS 3.20 ensures that floating-point errors are properly handled. IBM has also made a patch available for this problem. [editor's note; the patch installation information supplied by Microsoft has been deleted from this note]
>
> If you are running any version of DOS 3.20 other than IBM PC-DOS 3.20, contact your hardware manufacturer (OEM) to find out if the problem exists in your DOS 3.20 version."

Using Microsoft WORD with a Tandy DW-II

Fm MISOSYS, Inc: One of the reasons that we refrained from using Microsoft WORD until recently, was that we were not aware that we had a letter quality printer which could be used with our IBM PC. Finally, after deciding to start reading the manual, we noticed that the 3.1 upgrade information noted support for many Tandy printers, including the DMP series models 130, 200, 400, 430, 2100, 2100P, and 2200; as well as the DW II and IIB and DWP's 220 and 510. Now we have two DW II printers which we use for our letter quality work (until we acquire a laser printer which is next on the list of acquisitions). So we decided to plunge in and go to work with the DW II.

The first job was a revision to the Little Brother Installation manual for TRS-80. That was needed because we were in the process of revising the LB three disk set operating under TRSDOS 6.2 to a two disk set operating under either TRSDOS 6.2 or LS-DOS 6.3. The installation instructions were now more intricate; thus requiring a major revision to the manual.

Logical Systems used Word Perfect to prepare the LB manual. So our first task was to port the installation manual source file over to WORD. This was relatively easy; it was a two part process. WP didn't have a utility to convert it's files over to WORD. Neither did WORD have a utility to convert WP files to it. However WP's utility could convert WP files to DCA Revisable Form Text and WORD had another utility, RFTOWORD, which converted these DCA files to WORD. Thus, the two-stage process provided me with a source file usable under WORD. The revisions began.

Everything went great until I went to print out the document. Everything came out double spaced! It was strange, because I noticed that the character formatting enhancements, such as BOLDFACE and underlining, which WORD achieves on a daisy-wheel by by sending a carriage return and then overprinting, did not double space; if the printer was generating an automatic line feed on a carriage return, it should have double spaced there as well!

The manual indeed said that it supported a DW II. I also thought that was odd since I never knew that my DW II could do anything but generate a line feed automatically after a carriage return. I next turned to the DW II manual. Sure enough, on page 14, I observed that the control sequence ESCAPE-^U enabled carriage return only while ESCAPE-^V disabled carriage return only. The note attached to those operations stated, "normally code 13 causes Carriage Return + Line Feed. However after the code sequence of 27,21 (1B,15), input of 13 causes Carriage Return only; 27,22 (1B,16) causes return to normal." Therefore, there was a way to dynamically switch the printer to suppress the automatic line feed on carriage return. This then required a greater degree of analysis.

WORD has probably the greatest support of printers, or rather the support of the greatest number of printers. It comes supplied with oodles of printer drivers; but its documentation on modifying its printer drivers is designed for printer manufacturers, or those knowing a lot about their printers. WORD provides a utility, MAKEPRD, to convert printer drivers (PRD files) to WORD documentation files and vice versa. Thus my first step was to convert the DWII.PRD file to a .DOC file. This then can be edited by WORD (or at least examined).

I don't want this to develop into a tutorial on writing WORD printer drivers, so I'll get write to the point (I may do that in the future, since WORD doesn't have a DMP 500 PRD and that's one of my printers). A PRD is divided up into sections. There's a HEADER which describes the physical aspects of the printer; a FONT DESCRIPTION section which describes the various fonts supported by a printer (including the type wheels); a CHARACTER WIDTH TABLE which describes the width of each character needed for microspace justification; a CHARACTER TRANSLATION TABLE which can be used to create artificial characters by combining discrete characters; and finally the PRINTER CONTROL SEQUENCE DESCRIPTION. It is this latter section which I focused on.

The PCSD has a record described as, "Resets printer, beginning of document". It has another record described as, "Sets line spacing, second part". That is noted for Tandy printers with the following description: "Uses control sequence instead of carriage return to move print head to leftmost printing position (for multipass printing of boldface, underline, strikethrough)." Here's what the original PRD file had for those two items.

```
{P
byte:0 mod:0 "^[^0"
byte:24 mod:7 "^[^U^M^[^V"
}P
```

Pay attention to the strings. The "^[^ indicates the value of ESCAPE which is 1BH. If you examine an ASCII chart, you will see that the "[" has a character value of 5BH. The "^" is the standard nomenclature for indicating the <Ctrl> key. Since <Ctrl> strips all but the 5 low-order bits, "^[" designates a 1BH. Thus, that string specifies the sequence 1B,0F. The DWII manual notes that as setting 1/10" space node. Similarly, the byte:24 string generates the sequence, 1B,15,0D,1B,16. If you look at page 14 of the DWII manual, you will see that this sequence disables the line feed after return, sends a return, then re-enables line feed after return! That means the printer will always double space on lines which contain no emphasis formatting! That was certainly wrong. Here's how I re-coded those two sequences:

```
{P
byte:0 mod:0 "^[^0^[^U"
byte:24 mod:0
}P
```

With this change, my "resets printer" string generates the 1B,0F,1B,15 sequence to keep the automatic line feed on carriage return sup-

pressed for the entire document. It was a simple matter to then use the MAKEPRD utility to convert this revised .DOC file back to a PRD file. The printer spaced like a champ. I was in business!

Since the DWII.PRD file supplied by Microsoft with their package could not possibly work, I decided to advise them of my research. I received a reply to my letter rather quickly, which surprised me. Here is what they had to say:

> 'Thank you for your comments on the DWII printer driver. This printer driver was written and tested by Tandy. Actually, since they could not provide us with the DWII to test they assumed the responsibility of supporting any problems that come up with the DWII.
>
> However, a few weeks ago, I was working with a customer on a DWII problem which was the same double spacing problem when working from an IBM host. The modifications to the printer driver to fix this was similar to the ones that you recommend. The only thing different was byte:24. I kept the CR in (^M). I felt that I wanted to make sure a forced CR was sent in the event that incremental linespacing was needed. So my byte 24:"^M". Ann M. Marsh'

So there you have it. That's how to use your DWII printer with Microsoft WORD on a PC. Now, since I was able to use the DWII with WORD, I thought why not write a little .COM utility to generate that printer control sequence. Then I could use the DWII with other programs without having to re-prime the printer. Here's a little assembly language program that can be assembled with our ED/ASM-86 product:

```
;
; INITDW2.ASM - Program to condition a
; DW-II printer to suppress automatic
; line feed on carriage return. Must be
; invoked after the DW-II is powered up
; and connected to the PC's printer port
;
START:  MOV     AH,40H
;designate standard list device
        MOV     BX,4
;2 character string
        MOV     CX,2
;Point to the string
        MOV     DX,OFFSET NOLF
;Invoke the DOS call
        INT     21H
;init for error
        MOV     AL,1
        JC      ERROR
;Set to no error
        XOR     AL,AL
;Terminate & return to DOS
ERROR:  MOV     AH,4CH
        INT     21H
;cr only on 13d
NOLF:   DB      1BH,15H
        END     START
```

MISOSYS Products' Tidbits

Assembly: EDAS, MRAS

Fix a long-standing bug in FIXUP/CMD

Fm Walt Gabriel: Re: FIXUP command from MRAS. When I used the command line,

FIXUP CC/CCC (HEADER)

the program filled an entire disk; same with (NUMBER). Is there a patch?

Fm MISOSYS, Inc: Since you supplied the name and origin of the input file which you were having a problem with, I was able to easily substantiate your claim. The problem stemmed from some inaccurate code to fix up a NULL source line. The code had two unnecessary DEC HL instructions which were to "backspace" over the encoded line number. However, the FIXUP implementation didn't use an encoded line number [the code fragment originated from EDAS which did use an encoded line number].

The easy fix is to NOP out the extra instructions. The patches are MFX51/FIX and MFX61/FIX found in The Patch Corner. Sorry for the inconvenience. This is the first report of this difficulty where we have been able to verify a problem.

Getting started with assembly code

Fm Seth H. Barovick, DDS: When I read the documentation for PRO-CREATE, I was completely lost, especially the part where the writer expresses that this is not an instruction manual for machine language programming. I did not expect this, but I think it is only right that some instructions be given for converting documentation in a magazine to a COM file. I know that it is a relatively simple procedure, but it depends on the assembly program being used. If you could give me these simple instructions, I will appreciate it greatly.

Fm MISOSYS, Inc: Here's additional instructions for using our assembler package, PRO-CREATE. Assembly language is not something to take lightly; our manual targets a specific audience. Obviously, your use of the package doesn't fall into the realm of what we would expect from an assembly language programmer. Nevertheless, let me give you a few tips on getting started with copying programs from a magazine and converting them to a command file.

First, source code programs are assembled to a 'CMD' file, not a 'COM' file. That's a minor point, but one which needs to be clarified. The executable object files under CP/M and MS-DOS are termed 'COM' files; however, under TRS-80 TRSDOS-type operating systems, they are termed 'CMD' files. You will always be dealing with two files: a source language file is named progname/ASM and the assembled output file, properly called an object code file, named progname/CMD. The difference is significant; you don't want to overwrite the source code with the object code. In interpretive BASIC, you have only one file to deal with.

Next, assembly language statements have a structure to them we call syntax. This is discussed on page 2-5 of the manual - which is very clear on the point. If you are going to use the line editor which is internal to EDAS, you start it off by typing the command, "I", once you have invoked EDAS. This is very similar to the "AUTO" command of BASIC. In fact, the editor is quite similar to what you should already be familiar with in BASIC. Of course, the assembly language statements are entered into "fields" separated easily with the TAB key. The EDAS line editor follows the TRS-80 keyboard convention and uses the <RIGHT ARROW> key as TAB; SAID screen editor uses <CLEAR RIGHT ARROW> since <RIGHT ARROW> by itself is used for cursor movement.

Just follow the statements printed in the magazine. When you have entered all statements, depress the BREAK key. This returns you to the EDAS command prompt. Save your source file with a "W" command. Then assemble the file with the "A" command. Check the manual for the various options at your disposal. You most likely will have typing transcription errors. If you use an assemble command such as, "AWE", the assembler will pause every time it detects an error. When you have corrected all of your errors and re-saved your source file, assemble to disk with a command such as, "Aprogname-NL". This generates the 'CMD' file. You can obtain a printer listing of the assembled program by using the command, "A-LP". This can be combined during the CMD generation with, "Aprogname-LP". That should be enough to get you started.

Just what is relocatable?

Shane Dawalt To MISOSYS, Inc: I was running PRO-MRAS, ver 1.0a and got a "Relocatable Reference Error" message during the assembly. I quickly looked up what that error was and why I was getting it when I realized, it is not documented in the error message section. Did you know that? Anyway, it appears on one particular line:

```
DC    $&0FFH,0
```

This was "supposed" to provide a page break. I have a variable storage area I want to place on a page break so I can find it easier. This program is being assembled as a relocatable module (I have some library routines I want to link in). My problem is a simple one ... how do I force the PC to a page break? The solution is proving to be a bear. OH, and one piece of important information (perhaps); the line displayed above is in a DSEG area.

Now for another "questionable" question. I want to obtain a listing of the program after it's linked. I want (but may not necessarily get) a listing of all the labels, not just the public labels. It's a REAL problem to debug a a program with a listing showing the relocation addresses (not absolute) which was generate BEFORE MLINKing. Any ideas?

Fm MISOSYS: I don't know what you mean by a "page break" with code; but, on the other hand, the DC instruction generates BYTES. BYTES are not relocatable, only 16-bit words. Your operand field referenced the program counter which is a 16-bit relative field. That's why you got the error.

It's impossible for anything in MRAS to generate a "listing" of all the labels. In fact, symbols not declared GLOBAL (or PUBLIC) are considered local to the module being assembled; the linker doesn't even have any information on them. Best thing you can do is get listings of each module with relative symbol tables, then generate a public symbol table in the linker. You then have to use the public symbols of each routine as a base reference. Now if you get ambitious, you could generate symbol tables to a disk file then write a program to generate one long reference based on absolute addresses. That's not an impossible task, but it is not a trivial task.

Paul Bradshaw To Shane Dawalt: To force the PC to move to a new 256 byte page boundary (x'??00'), I've always used the following:

```
DC .HIGH.$.SHL.8-$+256,0
```

Note that this is taken directly from Roy's programming reference for PRO-WAM.

Fm MISOSYS: That's only suitable for EDAS or some other absolute code generating assembler.

Fm Shane Dawalt To MISOSYS, Inc: I thought the correct term was page break? Anyway, what I was trying to say (but obviously didn't) was I wanted to place a line into the code so that, when it was assembled, it would generate zeros up to the next page in memory. I assumed a "page" is an address where the lower 8 bits are of value 0 and the higher 8 bits can contain any value. I was trying to introduce enough zero bytes to clear the memory up to the next address where the lower 8 bits are 0.

As far as the idea on generating a symbol list, thanks for the idea. I don't think I'll get so ambitious as to generate a program to do what you proposed. I'll leave that to others more versed in relocatable assembly.

Finally, I'm sure MLINK has MLK68/FIX installed. Then again, this is the same person talking who swore up and down that MC was patched correctly too. I'll have to find my MISSING patched master backup disk and dig out the TMQs.

Fm MISOSYS: I usually refer to a "page origin" for what you wanted to accomplish. Indeed, after reading another message directed to you concerning what you wanted to accomplish, it

became clear to me [wanting to advance the program counter to the next address page boundary]. Maybe the week off fogged my brain. I believe, though, that my answer to you was still correct. You were trying to perform an 8-bit relocation. That is not supported by MRAS nor by MLINK. Actually, you were trying to do more than an eight bit relocation but using an 8-bit result. Since the PC is not going to be established until link time, there is no way to resolve that instruction within the assembler. The DC generates a byte value(s); however, since you were deriving it from an operation on the 16-bit PC which was relative, that was flagged as an error. There is a page in the manual [pg 2-15] which tells you what operations are permitted for relocation. There are very few. You can ADD, and you can perform limited subtraction.

What are acceptable file extension characters?

H. Brothers To MISOSYS, Inc: Before I figured out the mistake was mine this morning, the dog was hiding under the table. Seems that EDAS/MAS and XREF don't look at file names in quite the same way, and that I've been corrupted by MS/DOS and CP/M for too long.

Here's the scenario: I had an assembly language program with the line "*GET MACLIB.ASM" near the top. It assembled and ran perfectly. The I tried to get a cross-reference to the program and the fun began. I assembled the program again with the -xr switch; no problems. But when I ran XREF/CMD, it reported it was building a symbol table and then immediately reported a bad input file and aborted. I won't tell you the things I tried, nor the oaths I uttered until I realized that EDAS and MAS are able to handle the dot as a file extent separator (that's nice) but that XREF cannot. And since the first thing in the /REF file was the file name "MACLIB.ASM", XREF kept giving up. Once I changed the file name, in the original program, to "MACLIB/ASM" and created a new /REF file, everything went well.

I don't think this is a bug that needs to be patched, but maybe a word or two in TMQ will help keep someone else from scaring every dog and cat in the neighborhood.

Fm MISOSYS, Inc: I'm not sure that they were treating the "." as an extension separator. MC is the only package where we overtly permit "." as well as "/" to provide greater portability with UNIX and MS-DOS. Probably, they were terminating the parse on the first character which was not alphanumeric. XREF may have been testing something else. Yes, sounds like a note for a README file.

Why does MLINK abort?

Shane Dawalt To MISOSYS, Inc: I really like using the interactive mode of MLINK for linking C programs. This allows checking the status of the undefined labels before actually linking it into an executable file. Most handy instead of waiting for the entire linking process to conclude, and shortly thereafter, finding out one label is undefined.

Now, during interactive usage, PRO-MLINK doesn't exit to DOS when, say, a non-existent file is issued. BUT, for some reason, it does exit to DOS if a drivespec is left off. An example would be (at the MLINK interactive prompt):

```
? hcls:
```

MLINK immediately responds with a DOS extended error message. The extended error message given is:

```
** Error code = 32, Returns to X'3CEA'
** Illegal drive number
File = HCLS/REL:
Last SVC = 102, Returned to X'1A19'
```

This immediately exits to the DOS Ready prompt. I don't understand why a file not found error exits back to the '?' prompt but a bad drivespec abruptly exits to DOS.

Fm MISOSYS: MLINK specifically checks for a "File not found" error code returned by the @OPEN service call and treats that kind of error as a warning. Any other error is treated as fatal. I suppose that my original design specs considered "all others" as being significantly more serious.

Now if you want to treat all @OPEN errors as warnings, change the byte at X'2E14' from 0CAH to 0C3H in PRO-MLINK. A corresponding location in MLINK is X'5B09'.

When does MLINK zero DS regions?

Shane Dawalt To MISOSYS, Inc: I'm beginning to appreciate the power of PRO-MRAS. It is quite

a bit nicer than RS's Assembly Language Development System. With all that power, however comes a bit of confusion. I have two files which I have generated into /REL files. They are originally CSEG files with data areas intermixed within the code. The data areas are not declared as DSEG for reasons which will become apparent later. When I run them thru MLINK, the /CMD file is generated correctly, but there are zeros in the data areas. I defined the data areas with the DS op-code. The -Z switch on MLINK has no effect on these zeros. I assume (by the information provided for the -Z switch) that I have to declare these data areas as DSEG in order to get the zeros out of the generated /CMD file ... correct? The major problem, I don't understand how to use DSEG. Wouldn't DSEG place the data area at 0000? I want the data areas left right where they are, after the routines which use them. I'm not quite sure how to go about this though. Any help would be appreciated.

Fm MISOSYS: The manual states that DS's are filled with zeros UNLESS they appear in a DSEG. If you must leave the data area at a fixed spot relative to the subroutine the "data" belongs to, then you will have to leave them in the CSEG (why do you have that requirement?).

Starting a DSEG doesn't put the data at 0000; it puts the data at RELATIVE zero starting in the DATA SEGMENT, or immediately following the last byte of the 1st data segment. The linker sets up where the actual data origin will be based on either the LINK origin (defaults to 2600H/5200H) or physically specified by you if you use the -D switch override. Once you start a second segment type, you lose control over where the contents of the segment will be relative to the other segment except for the limited control over the actual segment origin provided in the linker. Unless you tell MLINK otherwise, each segment's link origin immediately follows the last module loaded. A DSEG will come first followed by a CSEG followed by any COMMONs. This sequence is what L80 does and I decided to keep the same sequence. I know this sounds as clear as mud. Perhaps this points to a QUARTERLY article. Actually, the same concepts must be understood in the MS-DOS arena as well - especially when considering EXE files which MUST have separate segments for CODE and STACK.

Bryan Headley To MISOSYS, Inc: Roy - Agreed. This is a problem that a lot of people have, understanding CSEG DSEG business. A lot of this traces down to a vague description in the old M80 manual. I'd like to see the article!

H. Brothers To Shane Dawalt: In addition to Roy's comments, my August column (should be arriving on doorstep any day now) has some explanation of CSEG & DSEG and how they work.

Shane Dawalt To MISOSYS, Inc: Thanks for the reply. My reasoning for keeping the data areas together with their respective subroutines is for easier debugging. If it weren't for debugging, I wouldn't care where the data areas were placed. I'm going to read Hardin's article (when 80 Micro arrives) and will see where that places my understanding of DSEGs and such. I can see by your reply that I'll have some rough time adapting to the "higher" microprocessor assembly, a la MS-DOS.

Fm MISOSYS: Not necessarily. The tough road to hoe for 80x86 is in understanding the segmentation scheme of the CPU and in how you have to code to adapt to it.

The hi-res graphics library and UNREL

Paul Bradshaw To MISOSYS, Inc: I have a question that you MIGHT be able to answer... Regarding UNREL and the GRPLIB hi-res graphics library. If I were using MRAS, would it be possible to just LINK in GRPLIB as-is (providing I knew the labels and register setups, of course)? If I used UNREL to create source code, would that be directly assembleable by MRAS?

Fm MISOSYS: You have to be able to use MLINK to access the GRPLIB since UNREL doesn't support any link items which MLINK doesn't support. Thus, GRPLIB had to have been written in ASM code. You, of curse, would have to know the entry labels, etc...

C Language: MC

Take MC to the MAX

Vincent Domeraski To MISOSYS, Inc: Will the same MC package run on the Max-80? Will I need a new assembler? (I have LC/EDAS)

Fm MISOSYS: MC will run on the MAX-80 a little better than the Model III since the MAX uses

less of high memory than a III - MC leaves very little heap area during compile time. Yes, you will need a new assembler. MC requires either MRAS (ours) or Microsoft's M80.

MC and portability of C source

Marc Nowell To all: I note that everyone seems to have a low opinion of the Alcor C Compiler for the Model 4. This is the only other language I ever purchased for the Model 4. I used it for about 8 keyboard hours and decided that I definitely wasn't cut out for this new language! I'm glad to hear that I wasn't using the best implementation to be had. I do a lot of work on MS-DOS machines now and have been looking forward to Turbo C from Borland. When and if it ever hits the stores and I get a chance to become familiar with it, I'd like to get Roy's full-featured Model 4 C compiler for portability. My question is this: Does anyone have any experience with porting programs from machine/compiler to machine/compiler? I'm currently entertaining the thought that I'll be able to write code on a Model 4 to run on an IBM PC and vice versa, given memory constraints. Is this feasible?

H. Brothers To Marc Nowell: I got Turbo C a couple of days ago. Haven't really had a chance to put it through its paces yet, but it seems to be a reasonably good implementation (better than Turbo Basic for serious work). There shouldn't be too much problem porting C source code from MC to any MS-DOS compiler, or vice versa, as long as you remember the difference in machine size. You'll probably have to do a little clean up on each end for things like filename conventions, but any good editor and a couple of macros could handle the whole thing automatically.

Making smaller object files

Jeffrey Kline To MISOSYS, Inc: I have ALCOR's C package for the model 4 and am finding that it is very much memory hungry on its programs (compiler, etc.). I also am finding that the end programs are way too large. Does your PRO MC produce true REL code in assembler that can result in a small /CMD file if the source code is a small program of sorts? Seems ALCOR will make a small 5 line source program into a 14 K file! Any help?

Fm MISOSYS: MC produces an ASM file which gets assembled by either MRAS or M80. The trick is making the library into very small modules. This we have done. The link process, although consumes more link time since it has to link together so many modules, still produces the smallest executable command file. There are additional ways to cut that file down further by proper use of #options. For instance, you can turn off I/O redirection (default is on) and save a bunch of code you may not want. You can also turn off dynamic allocation for standard file opening and cut another bunch (default is to use dynamic allocation which then brings in a bunch of alloc and related functions.

H. Brothers To Gary Phillips: But isn't that what a linker is supposed to do automatically? Any linker which says "Here, take it all whether you need it or not" is not really worthy of the name, it seems to me.

Fm MISOSYS: Hardin, it may depend on what a linker is being told. For instance, if you use an extern statement in M80 to declare a C-function external but never actually invoke that function, L80 will go merrily ahead and link in that never-used function from a library. That's because L80 doesn't bother to check whether an extern'd (that would by EXTRN in ASM) symbol was ever a part of a CHAIN EXTERNAL. On the other hand, MRAS/MLINK make that determination. I even supplied a switch in MLINK to override that suppression so you could force the loading of extern'd but unreferenced functions at your option. So, it all depends on how "smart" you want the linker to be given some "dumb" input. This whole situation is the reason why I cautioned our MC users using M80/L80 against blindly inserting a "#include <math.h>" as every math function is extern'd in that header. M80/L80 users would wind up loading the ENTIRE math library and subsequently most of the low-level math functions - a big set of code. MRAS/MLINK users wouldn't even have to blink. That's one of the fine points most folks wouldn't even realize.

The lighter side of C

jeff brenton To Jim Beard: well, let's see -- a language designed by Californians... "local variables" would be "Objects in my Personal Space" there would be keywords "Karma", "granola", "BMW", "Hang10", "ArfArf", "Owow", "SurfsUp", "tubular", "Do Lunch" loops, "Vals"

and "buzz". Results from running a program would depend on the mood of the computer, calculated via the Karma function. and the computer stops completely if SurfsUp==True!

Changing function return types

Richard Watkins: I am learning 'C' and I have a problem I don't know how to fix. When using the function getchar(), the value returned is an integer. How do I convert this integer to a character? I can convert a character to an integer but I don't know how to do the opposite. Can anyone help?

jeff brenton To Richard Watkins: To answer your question, you needn't bother - the conversion is automatic, using certain compiler- and machine-dependent rules, such as just keeping the low-order bits (if chars aren't signed) or other means. For example:

```
char c;int i;
c= 'a';
i=c;
i++;
c=i;
```

will leave the ASCII value of 'b' in both i and c, but c will only be 1 byte wide, while i will be 2 bytes on a z80 machine. Such automatic conversions are the source of many portability bugs, however!

You should only rarely use the char type for a non-array variable. It is quite rare in C to use a single char variable for anything but to store a temporary value, to later be used in a compare or appended to a string. since all of the "char" routines in C actually return an int or a pointer (since a char usually cannot contain the value 'EOF' and still return all of the characters in a given character set), and whenever you do a compare with a char, it is promoted to an int anyway, why bother with the extra conversion time, just to save one byte?

If you are interested in C with examples, over in Computer Language Forum, we have a C tutorial, originally intended for MS-DOS machines, in the files CTUTR1.TAU and CTUTR2.TAU, in DL 4. You can get there with GO CLMFORUM. Don't let TAU format scare you; you can break it apart with a text editor, if you have one that will handle the size of the TAU file, that is. there is no compression, just straight text. maybe it's time for me to compile an LDOS version of TAU, though, since such editors are a bit rare in this part of the computer world!

Tim Mann To Richard Watkins: A "character" in C is really an 8-bit integer. So if you assign what you get back from getchar() to a character variable, voila!--now it's a character. One exception--if you try to read past the end of file, getchar() returns the constant EOF, which is equal to -1. (That's why it's defined to return an integer rather than a character, so there is a value that isn't a character, that it can use to signal end of file.)

*CL access from C

Fm Richard Watkins: 'C' seems to be fairly complex language but it appears that one can learn and then advance. My next project is going to be files. I am writing a bulletin board program as sort of a hobby. The one I wrote and using now is in Alcor Multi-Basic but it is slow. Can anyone tell me how to get characters directly from the comm line. My reference tells about pointing to streams but I don't quite understand what you point to when accessing a device like the RS232 or the screen.

jeff brenton To Richard Watkins: If you have the commline driver installed, you can open *cl for input, and also open it for output (separate streams). You would be best off allowing the name of the commline device to be specified on the command line, and then trying to open it, in case someone wants to name it *SP or whatever.

In MC, the function checkc(stream) will check the 'file' stream to see if a character is available. When checkc() returns something other than FALSE or EOF, then you can use fgetc(stream) to read the character. This lets you do other things while waiting for either keyboard or serial port input.

Actually, C is a very SIMPLE language -- that's why it is so hard to learn!

Richard Watkins To ALL: Can anyone show me an example of how to open a file in 'C' to read and write to the RS232? I can't seem to find any good references or any code that does this.

Les Mikesell To Richard Watkins: Just open the file "*CL" and read or write to it (assuming you have set *cl to com/dvr).

Shane Dawalt To Richard Watkins: You simply use the fopen() function for byte I/O. For the filename, you would use the "*CL" device name. That's it ... unless Les has any other clarifications which I haven't seen (entirely possible!). I've used this method a number of times to output certain pieces of output to the printer like:

```
printer = fopen("*PR", "w");
```

Of course, printer must be defined as:

```
FILE *printer;
```

Fm MISOSYS, Inc: We published a little terminal program in NOTES FROM MISOSYS, Issue IV back in December 1984. It was submitted by Knute Johnson of Burbank, CA. Let me re-print here for your benefit, modified slightly for MC. Knute wrote, "I have really enjoyed the new additions to the standard library. The clearerr() and cleareof() have been extremely useful. I had planned to write a simple terminal program in assembler to use with Compuserve and some bulletin boards just to avoid using Radio Shack's cumbersome COMM program [note from me, Ugh!!!]. The new functions made it a lot easier to write it in LC. The listing is attached if you would like to use it in 'NOTES' or for whatever. To run the programs, install the comm line driver, use SETCOM to set up the RS232 parameters [note, Model I/III LDOS users just set the parms when invoking the RS232T driver], and then execute the program. To end the program press the BREAK key."

Here's CTERM

```
/*
   CTERM -  A simple Terminal Program in C
   Version 1.3 - 9 September 1984
   Written By: Knute Johnson
*/
#include stdio.h
#option ARGS OFF
#option REDIRECT OFF
#option MAXFILES 2
#define COMM_DEVSPEC "*cl"
#define LF 0x0a
main()
{  int c; FILE *com_in,*com_out;
   puts("\nCTERM - Version 1.3\n\n");
   if ((com_in = fopen(COMM_DEVSPEC,"r"))
== NULL)
       exit();
   if ((com_out = fopen(COMM_DEVSPEC,"w"))
== NULL)
       exit();
   option(O_KBECHO,0);
   while (TRUE)
   {   if (checkc(com_in))
           switch (c = getc(com_in))
           {   case EOF:
                     cleareof(com_in);
                     break;
               case LF:
                     break;
               default:
                     putchar(c);
                     break;
           }
       if (checkc(stdin))
       {   if ((c = getchar()) != EOF)
               putc(c,com_out);
           else
               break;
       }
   }
   fclose(com_out);
   fclose(com_in);
}
```

Here's more on opening the *PR device

Jerry Wagers To All: Either I am misreading something, or I have made an error in my programs, but I cannot seem to get fopen() to return NULL if the device (printer) is not online from the following routine.

```
#include "stdio/h"
#option INLIB
main()
{   FILE *infile,*lstout;
    if ((lstout = fopen("*PR","w")) == NULL)
    {
        printf("Printer is not available\n");
        exit(1);
    }
    else
    printf("Printer is online and ready\n");
}
```

This routine should print "Printer is not available", if your printer is not turned-on, if I read my 'C' doc's right. Did I miss something somewhere? I've tried it with MC, PRO-MC, LC, MIX-C, MICROSOFT C version 4, and half a dozen other versions and it doesn't work with any of them. It falls through the if loop and performs the else statement, regardless of the status of the printer. Can somebody please point me in the right direction here?

I thought fopen() returned a NULL if it attempted to open a nonexistent device. (eg: printer which isn't online? I guess I must have misread something somewhere. Will dig again through the docs; you know, those things that come with software that we use to hold yesterdays garbage!

LDOS Support To Jerry Wagers: Opening the printer device for output has nothing to do with testing the printer status for "OK". Once the device has been opened, you can test the port by attempting to send a character and waiting for the error/timeout. This would be the most portable way of doing the test. In Pro-MC, you can also check the port directly by linking in an assembler routine that uses @CTL to check the printer status.

Shane Dawalt To Jerry Wagers: The fopen() only returns an error if it cannot OPEN the file (or device). It doesn't TEST to see if the file (or device) and be read or written to.

LDOS Support To Jerry Wagers: Do note that there is a big difference between the device *PR not existing (which is what would cause the fopen to fail), and the *PR device existing, but there not being a printer hooked up.

Les Mikesell To Jerry Wagers: A non-existent device would mean a device whose name did not exist. Open doesn't check beyond the DCB.

C has no null expressions

Fm Shane Dawalt To Roy Soltoff: I just got done compiling (or an attempt thereof) a program which I had written for compiling under PRO-MC. I am running PRO-MC version 1.5b under LS-DOS 6.3. At any rate, I would like to point out a couple of interesting error messages generated and a problem encountered while using MC/CMD.

I had left out a constant within a header file which was being compiled. MC flagged it with an "Illegal constant expression" and a "Zero size illegal" error which I can understand. What I can't understand is the next error is displayed with "Missing ']'.". It wasn't missing! Here is the exact output generated by MC for this particular error:

```
RECIPE/H: line 25: error 10: Missing ']'
char line[LINE_LENGTH];
                      ^
```

As can be seen, the ']' is already in the errored statement and MC is pointing at the alleged missing bracket. This was done for more that just one of these types of errors (I'm consistent in making errors and MC is consistent in it's reporting format!

Next, if a line doesn't have a ";" at the end, why does MC display the next line to be evaluated and point to it's first character? I would think MC should display the offending line and point to the last+1 character where the ";" should go.

Finally, MC gave an error message which is not covered in the books. The error message was "FREE: BAD BLOCK X'99FB', LEN X'E5E1'" after which compiling was terminated. This also happened again after I had made some changes to the code. This happened both times just after the following error message was displayed:

```
EDITOR/CCC:2: line 443, calc_tabofs() + 7:
error 10: Missing ':'
tabofs==0?tabofs=flags.tab:;
                           ^
```

The second try gave the BAD BLOCK error message which follows: "FREE: BAD BLOCK X'03FC', LEN X'C784'". Is this a bug or is it common with the conditional operator?

Fm MISOSYS, Inc To Shane Dawalt: It's hard to say just what error MC should report due to the invalid syntax which resulted because LINE_LENGTH was undeclared. The statement was

in error. My hunch is that the opening bracket should have been followed by a constant expression; it wasn't. The identifier, LINE_LENGTH, which followed the "[" was probably interpreted as the next input stream which would be seen if you had no array entry and omitted the closing bracket. For instance, the statement,

```
char line[];
```

is certainly legal. I would suspect that MC interpreted your source statement in that manner since it did not have an array size; the right bracket was certainly missing!

The C language doesn't really have a good sense of "lines"; it is character stream oriented. Thus, a carriage return (newline) character is treated as whitespace, the same as spaces, tabs, line feeds, etc. The logical division in C is the statement terminator, the semicolon. If you omit the statement terminator, the compiler will continue to "see" characters in the source stream until it determines that a syntax error has occurred. Sometimes it is tough to pinpoint the exact location of the error; it really depends on interpretation of the stream of source code characters which precede and follow the omitted semicolon.

The bad block error is a RUNTIME error. That should not be happening when running MC; the compiler should not get a bad block error. I note that if I surround the first operand expression with parentheses, the bad block error disappears. That error is generated by the free() library command when you pass a pointer which is not in the chain of blocks which have been allocated by alloc(). Now the compiler still does not like that conditional, so my hunch is that it is trying to free up something which it thinks it allocated but which it didn't. I think it also should not be flagging that conditional as an error. A work around is to re-code it as an "if". For example, the statement,

```
if (!tabofs) tabofs=flags.tab;
```

is equivalent to what you were trying to accomplish.

I passed your problem along to Rich who came up with the easy fix. The corresponding patches, BADBL??/FIX, are in THE PATCH CORNER. Here's what Rich had to say on that issue.

Roy, Here are the fixes for the bug that Shane Dawalt discovered. Correcting the problem was simple; however, to find space for the patch I had to shorten the error message which read "Illegal floating point operation" to "Illegal fp operation". As for Shane's C syntax, the statement is incorrect no matter where one puts parentheses, and the compiler was right to complain. My analysis is as follows:

```
EDITOR/CCC:1: line 443,
calc_tabofs() + 7: error 10: Missing ':'
tabofs==0?tabofs=flags.tab:;
                ^
```

The assignment operator has LOWER precedence than the conditional operator; thus the '=' terminates the second expression for the conditional operator, and the compiler then expects a ':' to mark the start of the third expression. Placing parentheses around the assignment clears up this syntax error.

```
EDITOR/CCC:1: line 443,
calc_tabofs() + 7: error 70: Illegal
expression
(tabofs==0)?(tabofs=flags.tab):;
                              ^
```

Here the second set of parentheses allows the compiler to accept the assignment as a valid second expression for the conditional, but now MC is expecting a third expression. UNDER NO CIRCUMSTANCES MAY THIS BE OMITTED (there is no such thing as a "null" expression). By the way, the first set of parentheses is unnecessary. I reference K&R, p. 214.

I would not use a conditional expression in this context; it seems more sensible (and simpler) to me to use something like this:

```
if (tabofs == 0) tabofs = flags.tab;
```

Class conversions in C

Shane Dawalt To Les Mikesell: A small question on "C" using PRO-MC. If you were to say:

```
charvl = intvl;
```

where charvl is of class char and intvl is of class int, this would place the LSB of of intvl into charvl. On the other hand, if you were to say:

```
longvl = intvl;
```

where longvl is of class long and intvl is of class int, would this place intvl into longvl as the LSW (Least Significant Word)? It seems so, but I can't find anything which explicitly states this. I've been trying to get hold of a copy of K&R, but can't seem to find any 'round here, at least at the library. Anyhow, it seems as though the integer value should be converted to the long class. I've tried this, but the long value obtains a value of zero. Yet, if I do a "fake" arithmetic operation such as:

```
longvl = intvl * 11;
```

it converts the number beautifully. I'm aware that during arithmetical ops, lower classes are changed to upper classes before the operation occurs, so this wasn't really a surprise to me. I don't see why a simple assignment statement doesn't do the same thing?

jeff brenton To Shane Dawalt: An int, when made into a long, will be sign-extended into the long in a machine-dependant manor.

Les Mikesell To Shane Dawalt: In 'C', assignments perform the appropriate type conversions (to the type of variable receiving the value). Note that your concept of LSB/LSW does not necessarily apply to other machines but the values are made numerically equivalent within the range allowed by the types. The conversions of signed numbers may also differ among machines. Also, the conversions that are automatically done by the compiler for assignment or comparisons do not happen for values passed as parameters to a function.

Using the correct conversion character

Fm Richard Watkins: Here is a sample program showing my problem. Can anyone tell me why the following C program returns a value of zero no matter what the value that is entered?

```
#include stdio.h
#include math.h
#option BBSLIB
main()
{  char string[10];
   double num;
   puts("Enter number");
   gets(string);
   num=atod(string);
   printf("\n Number is: %d",num);
   exit(0);
}
```

I would appreciate any help that any of you can give me. I am trying to learn C but having a very difficult time at it.

Fm MISOSYS: Your answer is ZERO because you used "%d" in your printf() statement which expects an INTEGER argument. Your argument was a DOUBLE. Thus your printf() statement should use the "%lf", or "%lg" translation. Remember, the translation code MUST match the argument. You are not alone on this; it is quite easy for the beginner to consider the %d conversion specifier to indicate "double" when in fact it indicates "decimal". It is better to consider "doubles" as "floating point numbers"; that's where the "%f" comes from. The "%e" is easily remembered when you think of "exponent format". Got it?

Les Mikesell To Richard Watkins: The %d specifier means you are going to pass an integer for that field. For a float or double you should use %f, %e ,or %g specifiers.

Shane Dawalt To Richard Watkins: What Roy said!!? That same problem gave me kittens for an entire day. I wanted to kick myself when I realized the solution.

Here's some more on extern'd identifiers

Richard Watkins To all: While using PRO-MC how do you access a structure that has been declared as a global in another file? Is it correct to use 'extern' outside of the particular module in the second file? I was writing a program that has several functions that will make use of the same structure so I declared it global. I have been writing one function at a time and debugging it. To save compile time I work on each new function in a separate file and when I am satisfied I add it to a special library.

Shane Dawalt To Richard Watkins: Yes, you can use 'extern' as in:

```
extern struct xxxx yyyy;
```

xxxx is the structure tag (if you indeed have one) and yyyy is the identifier. You will have to compile the structure with any other module requiring access. This is because the module doesn't have any idea what each member of the structure is without the structure definition. I usually place the structure definition in a header file and use '#include MODULE.H' within each module which requires usage of the struc-

ture. BTW, I think it's easier to include the extern declaration within the header. BUT, if you do this, you cannot use the header file in the module containing the main() function. 'Cause this is where the structure storage is set aside. It would be quite confusing to the compiler to be told, "Ok, allocate memory here for the structure, but note that this structure is extern to this module!? (I've done it before too.) Clear as mud?

Richard Watkins To Shane Dawalt: I didn't know that I could put the structure in a header file. I didn't even know I could create my own header files. How do I go about creating a header file?

Shane Dawalt To Richard Watkins: Just tuck whatever needs to be given to all your modules into one file and place an extension of 'H' on it like MYPROG/H. In each module you require the data (some or part) in the header file, simply type "#include <myprog.h>". Of course, you don't have to use the "<>" if you don't want to, and you can use a "/" instead of a "." before the extension. But I'm so used to typing "." on the computers at work now that it's second nature (unfortunately). You must watch what you put into your header files? Don't put something like:

```
int some_int;
```

because every time the header is used, the compiler will create a NEW location for SOME_INT. I did this once and had a DEVIL of a time figuring out what was wrong. (Of course, when I figured it out I wanted to kick myself [but my foot wouldn't reach].)

A header file is GREAT to give extern references. You don't have to type them in EVERY module you create for your main(). You simply type the simple #include .... directive and that's that? Easier than keeping a listing with all the current external variables. (I wrote a program with has 5 modules to it. A header file was required for it. I wasn't about to retype the structure definition 4 times. I was lucky to type it once!)

On initializing bit fields

Shane Dawalt To MISOSYS, Inc:: Just got my K&R today, YIPPIE? Anyway, I've been having some problems with PRO-MC initializing static and external structures. I have dug back through my TMQs and have found nothing dealing with initialization of structures. When I attempt to initialize a structure (external or static external) OUTSIDE of a function, I get an "Illegal Initializer" error. If a STATIC structure is initialized within a function, everything is fine. The structure would looks like this (a carat was placed where the compiler gets angry):

```
struct {
   unsigne? ? f_outeo?   :1;
   unsigne? ? f_EJpag?   :1;
   unsigne? ? f_genpen? :1;
   unsigne? ? f_curvge? :1;
   } status = {0,1,0,0};
                ^
```

K&R says (as does the MC manual) that only automatic structures or variables cannot be initialized. External and static classes certainly don't fall in the automatic class. The compiler is the most current one v1.5b (with the patches provided within TMQ. BTW, this is the MC/CMD file, not MC1 & MC2 files.

jeff brenton To Shane Dawalt: I haven't ever seen anyone initialize bit fields before - could this be the problem? I will have to dig out some books to see if this is legal. [it is time to move my K&R and H&S over to the new bookshelf next to the computer, anyway...]

MISOSYS, Inc To Shane Dawalt: Shane, According to Rich, a field cannot be initialized in the same statement as its definition. K&R doesn't say you can't; but then K&R doesn't say you can. In fact, K&R is quite confusing on the point. On page 137, K&R state, "The syntax of field definition and access is based on structures." Notice that it doesn't say a field is a member of a structure. K&R go on to say that, "fields are referenced ... just like other structure members" to further confuse the issue. K&R state that "members of a union are accessed ... just as for structures" (page 139) and on page 198 they state that "it is not permitted to initialize unions or automatic aggregates." An "aggregate" is defined as a structure or array. I further checked all of my books and found no statement or example where a bit field was initialized within the statement declaring it. "C Primer Plus" by Waite, et al, show field declarations and then go on to initialize the field values by separate assignment statements (pp 502-503). "Introduction to C" by Chirlian discusses fields (referenced as a "modification of the structure") on pp 154-155 but also uses a separate assignment statement for initialization. Purdum in "C Programming Guide" says, "A com-

plete C compiler may or may not support bit fields."! Rich has a copy of the new System V specs (6-volume set) which includes one book covering the C language definition as implemented in AT&T System V.III C compiler? I'll let you know what he finds in that. For now, use a separate initialization statement as shown in all the books I have researched.

Shane Dawalt To MISOSYS, Inc: Thanks for the in-depth reply. And, just for the record, I would support the decision to provide bit field initialization (cause when you have 15 different flags, it gets a little nasty). I know, not all of them will be initialized to logic 1, but ...

Fm MISOSYS: You won't find a change in our compiler to support it. I also just got a hold of MSC 4.0. That does come with a great set of docs. They do have a very specific language definition, but they don't shed any light on field initialization. When I get a chance, I'll see if they support it by trying it out. Have you ever come across a book which gives a field example showing initialization with the declaration? Don't forget also, that the separate assignment statement could be done as:

```
s.f1=s.f2=s.f3=s.f4=1;
```

for how many fields are initialized to the same value. Rich also brought up the other subject that unless you are using a bit field for a large array, the code generated to access the field would far exceed any memory conserved by using the field over an int. The MSC example for bit fields used an array of bit fields for a screen buffer of 25x80. It may make sense there.

Shane Dawalt To MISOSYS, Inc: No, I haven't seen anything with initializes a bit field. And yes, I realize that s.f1=s.f2= ... =1 could be used. I haven't used that type of statement since I was taught that you don't do those sort of things when programming. (Old habits die hard you know?) Anyway, I'll certainly look into using the above statement and see how readily the code can be understood. Thanks for the idea.

MSC 4.0; don't tell me, let me guess -- Microsoft C ... correct?

Fm MISOSYS: Yes, MSC is Microsoft C. "MSC" is actually what you execute. Why were you taught not to initialize a group of variables to the same value with,

```
var1=var2=var3=,,,=varn=constexp;
```

since that construction is memory efficient and speed efficient? There are no side effects with that construct as long as you don't have any increment or decrement operators present.

Now I did get around to testing bit-field initialization using MSC with the following:

```
#include <stdio.h>
struct {
   unsigned  f_outeol      :1;
   unsigned  f_EJpage      :1;
   unsigned  f_genpens     :1;
   unsigned  f_curvgen     :1;
   } status = {0,1,0,0};
main()
{
printf("%d %d %d %d\n",status.f_outeol,
status.f_EJpage, status.f_genpens,
status.f_curvgen);
}
```

and MSC compiled it without complaint. Just to investigate further, I tried changing one of the values to be out of range with what would be valid in the field size specified in the declaration. Changing that first zero to a seven did not cause the compiler to complain, although I think it should have. It's true that C does not usually complain when an initialization value overflows the size of the variable; but those fields are a different matter.

A rename() function in C!

jeff brenton To all: I can't remember -- did anyone ever come up with a reliable rename() function for MC? it is not included in the MC library, but there are a number of programs, including one I am compiling for posting here and on Computer Language forum, that need it.

Fm MISOSYS: You could use the exec() function to issue a DOS RENAME command. Since no DOS service call exists in LDOS to RENAME a file/device, access to the library command is the only route (short of rewriting the system code to put in a rename service call - you could put it into a SYS13...).

diskDISK

Problems with diskDISK and Aerocomp 20Meg HD

Fm MISOSYS: The following dialog may be lengthy, but we wanted to convey the difficulties we go through in isolating a problem users are experiencing with one of our products. A copy of diskDISK is priced about $40, yet we have spent perhaps $200-$400 of time investigating this problem; only to discover that the cause of the problem was in another company's hard disk driver. Of course, our customer(s) also spent a great deal of time aiding in the search of the problem. It points out the level of detail we sometimes need to resolve a problem. So when we ask for specifics, you know why we are asking for details.

Fm Ron Ungashick To MISOSYS, Inc: My configuration only contains the drivers for the hard drive followed by a SYSTEM (SYSTEM=3). After booting and execution of the configuration my drives appear as follows for mod 3 LDOS 5.3:

:0 Fixed LDOS 5.3 SYSTEM
:1 Fixed LDOS 5.3 DATA
:2 Fixed LS-DOS 6.3 DATA
:3 Floppy LDOS 5.3 SYSTEM

I have formatted a diskDISK file on drive 0 using the mod 3 diskDISK DDFORM. I am attempting to activate the diskDISK file using the mod 3 diskDISK as follows: DD :7 DDTEST. I receive the appropriate message informing me that the diskDISK file is active(?) When I execute a FREE command drive :7 shows up but most of the information is missing. A DEVICE command shows drive 7 data garbled, taking up two lines. A DIR command hangs the system. If, before I set up the diskDISK, I swap the HD system drive with the floppy via SYSTEM (SYSTEM=3), which makes the floppy drive 0, I can enable the diskDISK file and it works as it should. I can then swap drive 0 back to the HD and the diskDISK file still works. (Note: I did not have to disable the diskDISK file, although for the mod 4 version I do need it disabled as stated in the Docs or I get garbage). So to sum it up, I am unable to set up a diskDISK file on HD which is currently serving as the drive 0. For this configuration is the only way possible to have diskDISK files on the HD partition serving as the system drive to use the floppy as the system until the diskDISK files are enabled on the HD, then swap drives via the SYSTEM command; That is the only way I have got it to work. It is a problem because every time I wish to switch to another diskDISK, I must swap the Drive 0 back to the floppy, enable the diskDISK, then swap back to hard drive. I hope this explains my problem a little more clearly. Any assistance you can provide will be greatly appreciated.

Fm MISOSYS: Ron, I am now more puzzled than ever concerning your problems with diskDISK and its use on your 0 drive. Here's my results. I was able to DDFORM a 40D1 diskDISK (that's 40 tracks, double density, 1-sided) on my system drive. I was also able to assign the diskDISK to drive 6 or drive 7 (the only two slots available) and had no trouble with FREE or DIR. I tried this out using both TRSHD3/DCT as the HD driver and RSHARD5/DCT as the HD driver. Thus, I found no trouble. Perhaps your problem stemmed from the type of diskDISk you formatted but your message did not give its details. You also stated that you were able to switch the zero drive between the floppy and the hard drive AFTER the diskDISK was assigned to the hard drive. I can't imagine how that would work since the assigned diskDISK references the host drive based on the drive number at the time the */DSK file is opened during assignment. All h*ll will break loose if you switch the host drive's DCT position. The manual specifically warns against this, "don't do anything which will ... modify .. a logical drive slot (DCT) that is currently assigned to a DiskDISK". Got any other info?

Fm Ron Ungashick To LDOS Support: The problem occurred when I was attempting to access the installed diskDISK drive. It should not need to be DATECONVed since I am creating it under LDOS 5.3 or at least I think that that is true.

Fm LDOS Support To Ron Ungashick: The diskDISK file will need to be DATECONVed if you wish extended date and time handling on the files stored there, no matter what system the diskDISK file is created with.

What are the exact parameters for the system partition? How many cylinders and how many heads? Is the double bit set? Is this a head-only partitioning, or cylinder partitioning? If cylinder partitioning, what is the offset, if any, from the start of the drive?

Fm Ron Ungashick To LDOS Support: I have an Aerocomp 20 meg HD. It has four heads and I

have divided into four partitions, one per head. The first partition contains the LS-DOS 6.3 system, the second contains 6.3 data, the third contains 5.3 data, and the fourth contains the LDOS 5.3 system. I will use DATECONV on them, although it does not appear to have anything to do with my problem.

Fm LDOS Support To Ron Ungashick: I agree that DATECONV should not affect the problem you are having, but will still be necessary to use the extended date and time formats within the diskDISKs.

I don't understand how you are formatting a 4-head, twenty meg drive for four logical drives and doing the partitioning only by head. A four head twenty meg drive has between 612 and 640 cylinders (typically). That would mean that each partition had one head and however many cylinders. But, you can't have more than 406 physical tracks per surface, so you would be losing at least two hundred tracks per partition this way. Are you sure that it is set up that way?

Fm Ron Ungashick To LDOS Support: I may have used incorrect terminology when describing my hard drive. I will try again. I have an Aerocomp 20 meg drive using Montezuma Micro drivers. I don't know if any other drivers will work with this drive. Each partition contains 153 cylinders. The only options I have when formatting are the head number and the number of partitions (1-4) for that head. I have mine divided by four for a total of 612 cylinders. I hope I am describing this better. I am new to hard drives and Aerocomp gave me next to nothing for documentation.

Fm Ron Ungashick To MISOSYS, Inc: You gave me an idea when you mentioned what type of diskDISK and I went back and tested some more. Under LDOS 5.3 on my hard drive system 0 a type 1 and a type 5 diskDISK drive works. Types 2 and 8 do not work. I also went back to LS-DOS 6.3 with the following results. Type 2 LS-diskDISK drive works and types 1, 5, and 8 do not work. Prior to this on LS-DOS 6.3 I was only working with type 2 so I incorrectly assumed that I had no problems with it. I hope you can duplicate some of this otherwise I think I'm going crazy. By now you may already think that. I was unable to duplicate swapping the system drive between the hard drive and the floppy. As you stated, all h*ll did break loose. The one day I was doing a lot of testing with the diskDISK drives this did "appear" to work but I could have screwed up the DCT's pretty bad by then. Maybe I am crazy. Anyway I hope this helps explain the problem I am having. Thanks for your assistance.

Fm MISOSYS: I'll try again with this new data. Perhaps it has to do with logging the disk. 6.3's @ckdrv operates a little different than 5.3's due to the fact that there is more room in 6.3's SYS2/SYS for a "larger" @ckdrv (that means more code to test things). That's only a hunch. But, boy, are those results strange.

Fm LDOS Support To Ron Ungashick: Well, unless I am misinterpreting what you say, that implies that they are doing quite a bit of logical mucking about with the physical structure of the drive. I'd guess that they are treating four physical tracks on a single surface of the drive as a cylinder. Not a very efficient way of doing things.

That also implies that you can't address the drive as less than four partitions, whereas in theory you should be able to do it in two.

Fm Steven Jerkins To MISOSYS, Inc: I'm looking at your letter dated 23 June 1987. Your reply about the problems I'm having with LS-DiskDISK don't help. I thought maybe I AM doing something wrong. I broke out the manual again, started with a new copy of LS-DOS 6.3, moved a copy for DD/CMD and DDFORM/CMD over to a new disk, and finally installed the hard drive. I DDFORMED a DSDD 80trk DiskDISK-- worked fine. Installed it as logical drive 7 -- worked fine. I thought "great" now I feel like a fool. I disabled :7 and DDFORMED a second DiskDISK -- DDFORM worked fine. I installed it as :7 and did a FREE command - utter trash. I tried a directory of :6 (the hard drive partition where I put the /DSK files. The directory on the hard drive was trashed. I re-formatted and re-loaded the hard drive from my backup floppies. I tried again. One time out of three, using the same commands, I get utter garbage for a DiskDISK.

Now, you mention needing line-by-line lists of commands that I'm using. That is no problem, I'll link *DO and a disk file for the session. You also said you need DCT dumps. That I need some instruction on how to do. I really want to get DiskDISK working for use with a BBS and I don't dare as it is behaving so erratically with my system.

Here is what I'm using.... 128K Gate-Array Model 4 2 Mitsubishi 80 trak DSDD external drives 20 Meg Aerocomp Hard Drive (Montezuma Micro Hardisk

Drivers) (It's an Adaptec when you look in the cabinet) ADC 1200B Modem Orchestra-90 Board (don't say it, the first thing I tried was disconnecting the ORCH board.... no change in Hard drive performance with DISKDISK) I have the hard drive in four partitions; one each for CP/M, LDOS 5.3, LS-DOS 6.3, and Bulletin Board The LDOS, LS-DOS, and BBS sections use the other LSI DOS partitions as datadisks. Let me know what info I can get you so I can figure out what's going on.

Fm MISOSYS: If you can give me a list of commands which will repeatedly duplicate your problem like the brief scenario you identified in your message, I'll start with that. But I need to know what you are doing before I can try what you are doing.

Fm LDOS Support To Steven Jerkins: As what you are doing should work fine with diskDISK, it sounds to me like one of three possible problems: (1) The drivers you got from Aerocomp are no good. (2) You have a bad copy of DD. (3) You have a hardware problem and/or the drivers are installed wrong. You are using "DD :n" with the proper parameter to disable the diskDISK drive, not SYSTEM (drive=n,disable), right?

I agree that the Orch 90 could be a possible source of problems. I'd leave it out of the system totally until all other problems have been resolved.

Fm Pete Granzeau To LDOS Support: I believe Orch-90 is a major problem when used, as it evidently turns off the 50-pin bus. It might be hard to read a HD that way.

Fm LDOS Support To Ray Pelzer: There is still the concern over contention on the EXTIOSEL* signal. Do you know how the Orch90 drives this pin in its hardware?

Fm Shane Dawalt To LDOS Support: Couldn't you just pull Orch 90 and if the bus works correctly, then note that the Orch 90 uses totem pole outputs instead of the required open collector outputs?

Fm Steven Jerkins To LDOS Support: The drivers from Aerocomp give no problem at all with any other program. Yes, I am using "DD :n (disable)". The same results come whether I have the Y-cable attached or not. I wish I could get a resolution on what is going on.

Fm Steven Jerkins To MISOSYS, Inc: I uploaded the file DISKPR.ARC. The file (archived with ARC4) contains a joblog of a DiskDISK installation and a listing of one of the files that was overwritten in the copying to the DiskDISK. This is an example of the problem I am having. I still need to know how to give you dumps of the DCT as you mentioned before.

Fm MISOSYS: You can go into extended DEBUG and PRINT memory from 470H. There's 80 bytes of DCT's. I can't imagine how you could have such a long JOBLOG which required ARCing.

Fm Steven Jerkins To MISOSYS, Inc: I did the ARC simply to get the joblog and the text file both together and to save time. I am so used to ARC'ng everything I get ready to upload I guess it has gotten to be quite a habit.

Fm MISOSYS: I got it off last night. I may be able to get to it before I go on vacation. However, finishing up this PRO-WAM Release 2 documentation takes top priority right now.

Fm LDOS Support To Steven Jerkins: All I can recommend is that you follow up with Roy on dumping the DCTs before DD, after the first DD and after the disable/enable of another DD.

Steven Jerkins To MISOSYS, Inc: I mailed a disk and the printouts of the DCT when using DiskDISK on Friday the 24th. The disk has a copy for DD/CMD and DDFORM/CMD and the hard disk driver to use the Aerocomp hard disk I have. (It's an Adaptec inside the case.) I hope this will aid in diagnosing the problem I am having and, hopefully, provide a solution.

Fm MISOSYS: We did dl your diskpr.arc file; however, due to pressing agenda, it will be a while before we get around to checking your scenario; after all, we did go through it once without any problem.

MISOSYS, Inc To Steven Jerkins: Steven, I have spent another four hours analyzing your diskDISK problem and now know why you are having the problem. Your Montezuma Micro hard disk driver is at fault. The values in the Drive Control Table for that drive are in conflict. Here's the scoop.

The DCT image you provided shows one of your HD partitions to contain: 32 sectors per track, 1 head, 4 granules per track, 16 sectors per granule, and the DBLBIT is set. Those figures don't add up. 4 granules at 16 sectors per granule are 64 sectors; yet, the SPT figure is 32! That drive is a 20Meg drive. I suspect that it has 4 heads. MM's driver probably internally groups TWO physical tracks as

one. Then the purpose of the DBLBIT is to show the system that two physical tracks equal one logical track. That drive probably has 612 cylinders. Since MM shows that drive to the system as 153; the DBLBIT can bring it to 306; they have to internally associate two tracks for every one that the system thinks exists in order to arrive at the 612. Now that drive then actually has 128 logical sectors per cylinder. This could be calculated as the SPG (16) times the GPT (4) doubled by the DBLBIT to equal 128. However, the SPC should also be capable of being computed from the sectors per track (32) times the number of heads (1) doubled by the DBLBIT is equal to 64!!! Since diskDISK calculates the number of sectors per cylinder by the latter method, it thinks that your drive has 64 sectors per cylinder instead of the 128 sectors per cylinder. Clearly, the data in the DCT is INVALID.

So what to do. I'll have to give it some additional thought. Since your hard disk driver is loading the DCT with incorrect data, it is at fault. It could be possible to circumvent the bad data be changing diskDISK's method of calculation. But then, some other program could give you the same kind of trouble.

LDOS Support To MISOSYS, Inc: Sounds like MM should be notified of the bug. I don't think that DD should be patched, I wonder how long it will take MM to come up with a patch or new version?

Fm MISOSYS: My first thought was, of course, that MM should fix their problem. Their driver would probably need a re-write as no reasonable patch could fix it. I suspect that a revision to use logical heads would be a better choice. They could show their internal double tracking by treating it as two heads. If they logically construct the entire drive as eight heads at 306 cylinders instead of 4 heads at 612, they could use the head count to correctly reflect 128 sectors per cylinder with both methods of calculation. Then the figures would be correct to the DOS. This also puts Steven in the middle.

Ron Ungashick To MISOSYS, Inc: Roy, A couple months ago I reported to you a problem I was having with both version of diskDisk. I also have a 20 Meg Aerocomp drive using the MM drivers. To refresh your memory, in short, some formats of diskDISK would not work. I am guessing now for the same reason as you just stated to Steven Jerkins. Do you sell any drivers that would work with the Aerocomp HD? I purchased both versions of diskDISK hoping to save disk space. Perhaps I chose the wrong HD. Have a good vacation!

Fm MISOSYS: I recollect the problems you reported and could not come up with a reasonable explanation then; nor could I reproduce your problems. Steven Jerkins was persistent in providing me sufficient detail concerning his operating environment. The clue came when I asked him for copies of his DCT. He also provided me a copy of the MM driver which I had to disassemble and interpret before I could pin the problem down to their driver. I would certainly expect that your problem stems from the MM driver, as well. You must be using their HD20AD4/DCT driver. DiskDISK absolutely will not work correctly with that driver because of the wrong information supplied to the system DCT relative to the drive's partitioning. We do not sell a driver for that drive. I don't think Powersoft does, either. Powersoft used to sell a driver for the older Aerocomp drive. That's when the controller they used was a Western Digital. That 20Meg controller is an Adaptec. I will probably be taking Joe's advice and registering a complaint with MM. That's where the trouble's origin is and that's where it should be corrected. Anything short of that is a band aid covering a hole in a dam.

LDOS Support To MISOSYS, Inc: I wonder how different the Adaptec controller is. If it isn't too bad, and Aerocomp isn't keen on fixing their code, your WD or Xebec drivers might be fairly easily modifiable. Hard to test without a drive though, and I don't think you need to pick up another Model 4 hard drive.

Fm MISOSYS: Joe, The Adaptec controller is what Lobo used in their WIN series of drives. I wrote that driver package. It really wouldn't take a whole lot of time to revamp the ??HARD package and produce one for the Aerocomp drives. Two things stop us. One, we make it a point not to support hardware where we cannot test out our implementation, and two we don't have the time to fool with it at this point. We are months behind on the DSM86 port which we have now begun.

Fm Ron Ungashick: I have a 20 Meg HD from Aerocomp with the Montezuma Micro driver. I phoned MM and they requested that I send my driver disks to them along with an explanation of the problem. I sent my diskettes along with Roy's description taken from the LDOS forum. They said they would resolve the problem. I sent my disks to them about a week and half

ago. I have not heard from them yet, but it is still early. I will post a message to the forum when I do hear from them. I have had no trouble with the drive itself, just the driver when using diskDISK.

Fm Gary Phillips: Please let us know if you get it [Aerocomp HD driver and diskDISK] working OK. I have decided to postpone my purchase from Aerocomp at least until I know whether this is a serious obstacle. After all, if I have to write my own driver, I might as well build my own hardware too and save a c-note or two! Or maybe I can still find a used RS unit somewhere...

Fm MISOSYS, Inc: The following letter was sent to Montezuma Micro:

> Attached is a copy of a letter which I have sent to one of my customers. The letter outlines what I believe to be the root cause of using our diskDISK product with Montezuma Micro's 20 Megabyte drive and HD20AD4 driver. The DCT data established by your driver is incorrect. The sectors per track entry multiplied by the number of heads doubled by the DBLBIT (64) does not equal the number of sectors per cylinder calculated from sectors per granule times granules per cylinder doubled by DBLBIT (128).
>
> Since your drive apparently has 612 cylinders and you wish to work around the DOS limit of 404 logical cylinders, might I recommend that you fix your driver by using a "logical number of heads"? If you treat that drive as 8 heads of 306 tracks per head instead of 4 heads of 612 tph, you will be able to work around the DOS limit and still keep the DCT figures accurate. You would then use a figure of four granules per cylinder in the DCT (where you presently use 8). In this way, calculating sectors per cylinder would be correct under both methods used by the DOS, its utilities, and other programs.

Hard drive users, do you need diskDISK?

Fm Marc Nowell: I recently acquired a 15-megger (Tandy) for my Model 4. I set it up as four logical devices so I could keep my three diskette drives and my RAM disk. My problem is that there is only 256 directory slots per logical device. I've heard of a MISOSYS package called DiskDisk. Is this what I need? Is it still available? What are the advantages and drawbacks of its use? Help!

LDOS Support To Marc Nowell: Yes, you need it. Yes, it is still available, $40. Advantages: works good and fixes the 254 user files problem. Disadvantages: you will probably want to re-format with two logical drives on the HD to gain back the logical drive slots to put diskDISKs into.

H. Brothers To Marc Nowell: I think a diskDisk is the only way to intelligently use a large hard disk on a Model 4. It works essentially like this:

You use a program called DDFORM to create a file on your hard disk that has all of the logical properties of disk: boot sector, directory, etc. There is a great deal of flexibility in creating this file. Then, when you want to access that logical disk, you run DD/CMD, which installs a driver and some information about the disk and opens the file. It also assigns the diskdisk to one of the 8 drive slots that the dos can handle. You can reassign the drive slot to a new logical disk, have several available at one time, etc.

There are only two disadvantages that I can see: First, you have to run Dateconv on a diskdisk when you first create it if you want time stamping under 6.3. Second, some memory is required for the driver (it can be either high or low) and a small amount of extra memory is required for each active diskdisk.

You might want to reformat your hard disk once you have diskdisk into one or two logical drives. The only files on one might be a list of diskdisks -- the other could have your boot-up stuff.

The diskdisk system is not quite as handy as subdirectories under MS-DOS, but it is close. If you make the diskdisks with the same size, sides, etc. as your floppy drives, you can do a mirror image backup of each as part of your backup procedure.

Marc Nowell To LDOS Support: Thanks, as usual, for the speedy reply. I'll check dl0 again! I also got a message from Hardin re: DiskDisk. Question: How many directory slots is in a DiskDisk? 256, like any other device? Can DiskDisk be run from JCL, allowing me to set up whole new batches of drives from a script?

LDOS Support To Marc Nowell: A diskDISK can be configured in many different formats. You can have from 36 to 254 available directory slots, with allocation from one sector per gran on up. Of course, there are trade offs between the various options, but I think you'll find it more flexible than anything you'll need.

diskDISKs may be formatted, installed, replaced or disabled from DOS or JCL, and may always be replaced or disabled from anything that allows DOS commands. Installing a new one in a DCT would require that one had been previously installed at DOS to allocate the linkage memory and then disabled. diskDISK can then re-use the allocated space.

Bob Haynes To Marc Nowell: I use (2) 15 meggers, and find DiskDISK indispensable! FYI, my HD is set up this way:

Drive	Heads	Use
0 HD	1	DOS; utilities; all primary programs; work space
1		disabled; Reserved for /DSK drive assignments
2 HD	2/3	DiskDISK files
3 HD	4/5	ditto
4 HD	2/3	ditto (secondary drive)
5 HD	4/5	ditto (secondary drive)
6		physical floppy drive 0
7		physical floppy drive 1

(Unassigned HD heads are used under LDOS 5.1.4)

Using the DOCONFIG utility from MISOSYS allows me to have any number of configurations quickly available, with different drives enabled/disabled, different KSM setups, and so forth.

DiskDISK is excellent not only because of the additional directory slots, but because of the organization it provides. When working on a project, who wants a directory of EVERY file on-line? By using DD, related files can be grouped together into /DSKs, and your directory becomes an INDEX of file groups, not the individual files themselves. You see only those files you are interested in, with much less confusion; and you can have multiple /DSKs on-line at one time, also. DiskDISK is very considerate of memory, requiring only 216 bytes of high driver memory. It also requires 42 bytes of low driver memory for each active /DSK. It reclaims used memory when disabled. Another thing: /DSK files can store data in smaller grans (1.5K for ss/dd format).

DiskDISK is available for both 5.x and 6.x; the programs are different, of course, but the /DSK files they create are identical and totally compatible. This might be very important to you if you access common drives between the two DOSes.

I'd also strongly recommend FM for your new HD; it's equally indispensable.

LDOS Support To Bob Haynes: One comment, all parts of diskDISK can go high, depending on available memory. Available memory in the low memory I/O driver zone is not required.

Pete Granzeau To Marc Nowell: Set up your three floppies as the last three drives in the table, and use DiskDISK (or is it DISKDisk?) as drive :4. It is well worth the expense and time, gives the effect of subdirectories in MeSsDOS, and also gives the ability to make rapid (mirror image) backups. Anything you don't need online all the time would be a prime candidate.

Bob Haynes To LDOS Support: That makes sense; my low memory requirements have been small enough that DD always had room to put the $DL modules there. Nice to know, though.

LS-diskDISK and the DOS network flag

Fm MISOSYS, Inc: The current release of LS-diskDISK does not close the /DSK file after it initially opens it to extract the directory information. This can be a problem if you have established the TRSDOS or LS-DOS network flag to be in an ON state. That flag keeps the file access system operating as it did in TRSDOS 6.0 and 6.1 - when a file is opened, a flag is set in its directory so that subsequent opens will be forced to READ ONLY. That feature was designed into TRSDOS 6.0 to guard against simultaneous file updating from two or more sources, conceptually called "network" operation. However, because programs were persistent in not closing files (as was permitted under Model III mode operating systems), the default operation of TRSDOS 6.2 was changed to inhibit the "file already open" bit in the directory unless the NETWORK FLAG was turned on. Here are some comments and two solutions from Bob Haynes.

Bob Haynes To MISOSYS, Inc: Roy, I had originally intended to post this for all, but decided since your products were exclusively involved, it should go to you. Post if you

wish, publish in TMQ, or use the cylindrical file. It works for me. I added the source for your convenience.

Are you running a Hard Disk system in conjunction with the PROWAM and DiskDISK utilities from MISOSYS? Have you set your system's Network flag to protect against recursive writes per the README/TXT instructions? Then you have probably noticed LOTS of spurious '?' open file symbols cluttering up your HD directories!

They result from the fact that some programs, because they are 'read-only', don't bother to close accessed files after opening them, and DiskDISK falls into this category. I prefer the '?' symbols to indicate only REAL problems, so I decided to modify DD to close those files properly. Here's the patch:

```
PATCH DD/CMD (X'260D'=CD A4 2E)
PATCH DD/CMD (X'2EA4'=11 1D 2D 3E 3C EF 28
05 FE 26 C2 80 2D 21 00 00 C9)
```

```
        ORG  2600H
START   LD   (STACK),SP     ;save stack
        CALL GETFILE        ;parse parms,
                            ; open file
        CALL Z,ENABLE       ;enable
        CALL NZ,DISABLE     ; or disable
        CALL NWCODE         ;formerly LD HL,0
        RET
;
;     new code at end of program
        ORG  2EA4H
NWCODE  LD   DE,FCB         ;get fcb
        SVC  @CLOSE         ;close file
        JR   Z,OK           ;go if successful
        CP   38             ;ignore "file not
                            ; open" error
        JP   NZ,ERROR       ;real abort (ERROR
                            ; is a DD routine)
OK      LD   HL,0           ; or normal exit
        RET
```

Fm MISOSYS: Actually, I think another way would suffice. I recollect that diskDISK doesn't use the FCB info except for getting the drive/DEC so it can get at the directory record of the /DSK file. Then it rakes off the directory extent data and saves it in the resident header. Thus, it should be possible for the "force read-only" flag to be set when the /DSK file is opened. That also saves you from having to CLOSE the file. I'll double check this. One way or another, the next TMQ will probably have something. Thanks for the input.

Bob Haynes To MISOSYS, Inc: Right on the nose, Roy; I hadn't considered that. Try this instead:

```
        ORG  2872H
OLD     LD   HL,BUFFER      ;point to buffer
        LD   B,L            ;LRL=0
        CALL @@OPEN         ;open the file
        JR   Z,$18          ;if no error
        CP   29H            ;'file open' error
        JR   Z,$18
        CP   2AH            ;'LRL fault' error
        JR   NZ,ERROR       ; else abort
$18     LD   BC,(DRIVE)     ;requested drive #
;
        ORG  2872H
NEW     LD   HL,BUFFER      ;point to buffer
        LD   B,L            ;LRL=0
; patch starts here at 2876H
        SVC  @FLAGS         ;IY to flag table
        SET  0,(IY+18)      ;Set read only,
                            ; ignore LRL errs
        CALL @@OPEN
        JP   NZ,ERROR       ;ng, abort
        NOP
; patch ends here
        LD   BC,(DRIVE)     ;requested drive #
```

```
.DD6BH2/FIX - patch to LS-diskDISK
D02,2A=3E 65 EF FD CB 12 C6 CD 4C 2E C2 80
2D 00
F02,2A=CD 4C 2E 28 09 FE 29 28 05 FE 2A C2
80 2D
. Eop
```

My understanding of the flag is that since we're forcing read only, no 'open' or 'LRL' errors will be generated. Is that assumption correct? And I believe you prefer the "D" type patch, also. Comments?

Fm MISOSYS: That is correct. The force-to-read-only SFLAG$ setting will also cause @OPEN to ignore LRL open fault as well as file already open conditions. It is also true that I prefer D-type patches when they can be tweaked in. The reason is a greater protection against patching the wrong file (or the wrong version). I'll check into your latest for the next TMQ. Thanks.

BASIC: EnhComp

Using EnhComp for 100% Z80 code

Fm Bob Schindler To Anyone: I am having a problem with PRO-EnhComp that probably is a cockpit error, not a bug in the system (I hope). I have absolutely ZERO trouble compiling BASIC and BASIC/ASM combination files. However, on a strictly ASM file, I am getting strange "Undefined label or symbol" errors at the very end of the pass. I have tried the '*Z-80' and 'Z80-MODE' directives in the file as well as the 'Z80' directive on the BC command line. The compiler lists about eight labels with this error. @@HIMEM is one that sticks in my mind. This problem does not seem to cause runtime problems (so far), but it is an aggravation when I am working under the 'S'upervisor, which is my normal modus operandi, since an interactive compile will abort the run on any compile error. I only tried the ASM only files to get some smallish programs out of magazines, etc. Except for this, I have NO complaints about the system. It was well worth the money (small plug).

While I was looking at the printout, I noticed another strange thing! Although the file contains an 'ORG 3000H' statement, the compiler shows the program loading at 2600H. Is this a bug or am I missing something? It is no problem to put the start address in the BC command line, but I am screwed under S/CMD.

Fm MISOSYS, Inc: re: your difficulty in using the global Z80 mode of EnhComp. You may not like this, but here's what I found out with my investigation.

In the discussion of compiler directives on page 2-4 of the EnhComp manual, it states that the '(G)' nomenclature references directives which are global while '(B)' is used to designate directives which are both global and local. The 'Z80' directive shows up with a '(G)' which indicates global only. That means it is only useful when entered on the BC command line. I confirmed this operation by examining the compiler's source code in addition to testing.

The symbols which were flagged as "undefined" when you tried to use the Z80 directive locally are referenced by a piece of code generated by the compiler before reading of the source file takes place. This is the BASIC initialization code. It is impossible to suppress this by a local directive since it is generated before source is read. That's why the Z80 directive is global only.

On the other issue of the program loading address, I believe you were mistaken. The ORG was perfectly successful. The program's load origin was indeed 3000H as you coded. Unfortunately, the only control you have over the START address is via the compiler command line. That's where the 2600 came from - it was the compiled program's START or transfer address. Typically, a transfer address is specified as an operand to an END statement. But EnhComp doesn't support this as it's primary use is a BASIC compiler.

Therefore, you have two reasons why you will need to compile the source file directly at the BC command level. One reason is to specify the '-Z80' switch; the other is to specify the START address which relates to the transfer address of your program.

Problem with search and replace

David Huelsmann To Fm MISOSYS, Inc: I can't find where this problem has been reported before. In PRO-EnhComp's CED, there appears to be a bug in the "X" search and replace function. The following screen output was captured thru PRO-WAM and accurately reflects the problem of truncation/dropping of portions of the code. x/tab/fn tb$. Have you seen this one before?

Fm MISOSYS, Inc: After looking at your code stream off-line, I noticed something very peculiar. Everything in the source line from the slash "/" character to the end of the line was truncated. The slash is also used to separate the search&replace command's strings.

After a very quick test, the problem was localized. Here's how that command works. It parses the command line into the search string and the replacement string. It then searches the source until the "search" string is found. That develops a pointer to the start of the line where the string was found [PL], a pointer to the first character of the search string in the line where it appears [PS], and a pointer to the first character in the line past the search string [PT]. The command then populates a local line buffer with the source line until [PL] is equal to [PS]. Next, the

command transfers the replace string. To do this, it uses a routine which terminates on either a NULL (the end of line terminator) or a slash (the end of string terminator in the command line). It then is supposed to transfer the trailing portion of the source line from [PT] until the end of the line. Unfortunately, CED uses the same routine as that which it uses to transfer the replacement string; thus, if the trailing portion of the source line contains a slash, as yours did, the transfer terminates prematurely. In order to solve the problem, that last transfer must use a routine which terminates only on end-of-line.

I have worked up two patches: one for Model III EnhComp, and one for Model 4 PRO-EnhComp. They are CED52/FIX and CED63/FIX respectively. Look for them in The Patch Corner.

Using Z80 labels for GOSUBs

David Huelsmann To MISOSYS, Inc: I have been struggling with PRO-EnhComp trying to get a Z80 in-line code segment to work. It is similar in concept to your example on page 6-1 of the manual. Essentially, this segment checks for a space at 0,0 on the screen, if a space is present, it writes a character to that position, if a character is present, it writes a space. I can get this code to work fine if I don't use a FOR NEXT construct or even a REPEAT UNTIL so I know that there is nothing wrong with the Z80 code. When I use the conditional expressions, I get a variety of error messages (RUNTIME) depending on just how I construct the conditional. The most common error seen is 0 (Next without For), though, I have also gotten a number of undefined error codes. Do you see anything obvious here?

```
CLEAR 400:TRON:DEFINT I,Z
CLS
ZB = ASC(":")
FOR I = 1 TO 1000: GOSUB "SYMBOL": NEXT
END
Z80-MODE
"SYMBOL"
LD A,15: LD B,1: LD H,00H: LD L,00H
RST 28H
CP 32
JR NZ,NONE
LD A,(&(ZB)): LD C,A: LD A,15: LD B,2
RST 28H
JR DONE
"NONE"
LD A,15: LD B,2: LD C,32
RST 28H
"DONE"
HIGH-MODE
'Perform a number of tasks
RETURN
END
```

Fm Phil Oliver To David Huelsmann: Something you should look at: I think the definition of "SYMBOL" should occur prior to the Z80-MODE statement. This is due to the fact that the usual compiled output is compact pseudo-code, for the most part. The Z80-MODE statement sets up proper execution of "pure" machine code. The GOSUB is a high level statement and thus when execution is routed to SYMBOL, the pseudo-code executor should be expecting pseudo-code. What it will get is Z80 opcodes, with the predictable unpredictable results.

David Huelsmann To Phil Oliver: Absolutely correct. Moved the "SYMBOL" label prior to the Z80-MODE statement and all executes correctly. I knew it had to be something I was just not picking up on. Appreciate the support.

David Huelsmann To MISOSYS, Inc: Phil was correct. The example shown on 6-1 shows a label after the Z80-MODE that is called within the HIGH-MODE. Might want to throw in a little correction in your next readme file. Got tmq I.iv today. Again kudos to you for an excellent issue. As one of 717 subscribers I would have to say the other 17000+ are really missing out.

Fm MISOSYS, Inc: Actually, that example worked as it was written. It may have been a fluke. I ran across your problem elsewhere with somebody else, though. Phil has shed light on the exact cause. I'll be extracting that for the README file.

Data Management: LB

Little Brother and Data Importing

Harry Hopkins To MISOSYS, Inc: I'm a new owner of LB86. It's great as a flat file DBMS at the price! I am having great difficulty importing a 48,000 name mailing list into it tough. The problem appears to be with the characters that are added to the end of the .LB data file. I've tried everything I can think of to import the data but keep running up against the "MISMATCH BETWEEN DATA AND DEFINITION" message. What's the best way to import to LB86?

Fm MISOSYS, Inc: The ONLY way without getting too pokey with file editors is to set up the data as a AUTO add function. You are a TMQ subscriber, so why not just read the article in TMQ I.iii. Page 74 has a BASIC program which can be used as a base for you to put your data into an AUTO job form.

The error message you quoted "Mismatch between data and definition files" gets generated when the record count listed in the data file differs from that in the data file. LB keeps the record count in two spots. If you have correctly pumped in data using some other means, just make sure that the record count in both spots agrees. The data file keeps the count in the record position immediately past the last record. If you have a file editor for MS-DOS, you can examine both of those files. Here's a good use for DED86.

LDOS Support To Harry Hopkins: The best way to import into Little Brother (either M4 or MS-DOS versions) is to convert the input to a JOB format file, with the necessary characters preceding to get into input mode. Don't forget to pre-allocate enough empty records.

Harry Hopkins To MISOSYS, Inc: Thanks. I've moved the data over from my Model 4 Profile+ using a segmentation program and TRSCROSS. Now I'm trying to get LB86 to pick it up. You mentioned TWO record counters; one being at the end of the data ("SQ.LB") file; but, where is the other count located? Failing this, which would seem to be the easier way to get the data imported, I'll try the JOB method. Thanks also for the reference to TMQ. I'll dig out my copy.

Fm MISOSYS, Inc: The other record count is in the data definition file "database/DEF". The DEF one is a little redundant and LSI suggested that I patch out its use. Because of that, you can get into trouble in restoring a data base with an old data definition file. The only "old" thing about it would be the record count it stores.

LDOS Support To Harry Hopkins: Dredging back in the dim recesses of antiquity, I think that this is right. The other is, unfortunately, at the beginning of the data file itself. There are six bytes, I believe, two for LRL, two for total number of records allocated and two for last record actually used. The problem is that these are not part of a whole "dummy" record, so you can't just skip the first record, you must skip the first six bytes. For some languages, e.g. BASIC, this is going to be rough.

Using LB under LS-DOS 6.3

If you are currently using the three-disk set of Little Brother in a 2-drive floppy 128K environment and have upgraded to LS-DOS 6.3, note that the start-up procedure supplied by LB automatically switches to a MemDISK which is loaded with a TRSDOS 6.2 system image. To run under 6.3 using this method, you need to upgrade to the current LB two-disk set. The charge is $12 + $2 S&H (US) plus your old diskettes. That gets you two revised disks plus a new installation manual.

PRO-WAM

Minor problem with PHRASE

Walt Gabriel To MISOSYS, Inc: You have a great package. Job is well done with the manual. Will the PRO-WAM toolkit be more specific on writing /APP programs completely with C?

[Here's two problems]: PHRASE exports the time. Three possible workarounds are, (1) turn off time display, (2) ignore, (3) erase with word processor. I am also unable to use DIALER with MODEM80 software when I uses Compuserve or a bulletin board. My workaround is to use KSM to dial the modem.

MISOSYS, Inc To Walt Gabriel: The first problem is an easy fix - easier than the

workaround you suggested. Rather than (1) turn off the time display, (2) ignore the time, or (3) erase with a word processor, it is easier and more effective to just relocate the PHRASE window. That's easily done with a one byte patch as follows:

PATCH WAM1/APL (D53,06=01:F53,06=00)

That's all it takes. This moves the window down one row. Obviously, No one ever tested PHRASE with the time clock turned on (or invoked from BRINGUP or TODO). I am making this a permanent change. Due to another small revision to PROWAM/CMD, I will be sending out a new disk soon to all recipients of 2.00b.

As far as DIALER not working with MODEM80, you provide me little information to check that out. On the other hand, I can guess at the cause. DIALER requires use of the *CL driver. Since MODEM80 can work through that driver, I suspect that they have trapped its I/O. That's my hunch. I leave it to you to confirm.

Displaying banks > 0 with MED

J. Verhelst To MISOSYS, Inc: I use your program MED/APP and with the "B" option, I do not see any difference on the screen, ";" changes the pages but not the bank choice. I have 1 meg option on the board. Do you think it would be possible to look at the banks above 3?

I also use the ALTDISK/CMD. Would it be possible to format the RAMDISK to up to 1 meg? For modifications if it helps, I may use PRO-CREATE.

Fm MISOSYS, Inc: The "B" option of MED allows you to display the contents of any bank page accessible through the DOS. If your "1 meg option on the board" includes a software interface to the @BANK svc of the DOS, then it is accessible to MED. Assuming that, don't forget that banks are switched in the address range 8000H-OFFFFH. Thus, you won't observe any screen change unless you are displaying a page in that range.

ALTDISK supports only a 32K or 64K ramdisk; there are no plans to change that. If you care to disassemble it and adapt it to your needs, that certainly is an option open to you. An easier method may be to write a ramdisk driver which interfaces to our @EXMEM svc presented in THE MISOSYS QUARTERLY, Volume I, issue iii (again assuming your 1 Meg board is known to the DOS).

TERM/APP: how smart is it?

Peter Amschel To MISOSYS, Inc: It has been a while since I tried to run the Term program. I have superdrive set up as drive 0 and need a terminal program to send files from the office model 4 to the model 4 with the BBS on it. In trying to set up the TERM program though, I failed and got that error message. I think I tried to set up @CL, or something like that first, and that is where the error message came in so I figured that I was not going to be able to do it with the WAM terminal program. Your assistance in setting up the WAM terminal program would be appreciated. Can it load a file from disk and then transmit it? I did not see any commands for it to do so in your documentation.

Fm MISOSYS: Sounds like you got the "No more room" message from whatever you were running when you tried to install the COM driver. That must be done from DOS Ready. That's the first problem. The second is that TERM/APP is a REALLY DUMB terminal program. It allows you keyboard and video. There is absolutely no file transfer capability, no printer capability, no protocol, no nothing. Thus, if you need file transfer, then TERM won't do it. On the other hand, if you need XMODEM file transfer, the XFTS package (which we used to sell but is now available bundled only in the MARK IV Collection) can be run as a library command via the LIBEXEC function of PRO-WAM. That's the best I can do. But you do have to install the *CL driver [SET *CL COM] first.

LDOS Support To MISOSYS, Inc: There is an XMODEM Pro-WAM application here in DL 2, I believe.

Fm MISOSYS, Inc: Yes, there is an X-modem PRO-WAM /APP on the forum. Forgot about that one.

SuperScripsit, Scripsit-PRO, and CTL-255

Lynn Sherman To MISOSYS, Inc: I received the latest version of SUPERSCRIPSIT yesterday. I will be looking to see what is needed to allow access to PRO-WAM from it. I'll let you know what I find out.

My office obtained SCRIPSIT-PRO quite some time ago. I patched it to allow PRO-WAM access only to find that constant crashes were the result. It turns out that SCRIPSIT-PRO makes indiscriminant use of the upper banks and clobbers PRO-WAM in the process. It may be possible for someone with extended ram to load PRO-WAM in a higher bank and access it from SCRIPSIT-PRO.

Fm MISOSYS: Thanks for the offer. I can't believe that SCRIPSIT PRO would use a bank which was unavailable as determined from the DOS SVC. Even knowing how Tandy operates, I can't believe that. But thanks for your efforts. We have gotten a few calls from SCRIPSIT PRO users about PRO-WAM interfacing but we have just told them can't be done. Actually, another method just struck me. What SS used was a @CTL call to the *KI device with function 255 (that was a special thing done for Tandy to keep them from releasing SS under their own version of the DOS). That function call does the matrix scan and returns an image of the scan in to the area pointed to by regIY. Now a simple filter could be installed after PRO-WAM is installed to trap the @CTL call (i.e. check for CTL function 255) then do a regular GET function. The trick is to have the filter know what the PRO-WAM activation code is. If the PRO-WAM activation was not in effect, then the special CTL function would be used. It may be worth a more fuller investigation. Of course, a larger secondary filter could certainly be done to even support export by taking the characters and rebuilding the scan matrix. Anything is possible.

Lynn Sherman To MISOSYS, Inc: I have a patch working for the new release of SUPERSCRIPSIT. I'll be uploading it here after I've tested it for a few days. I also have a patch that eliminates the delta symbols that SUPER-SCRIPSIT uses to represent two consecutive spaces.

I like your idea of a filter that could support export. I'll have to give some thought to it. May be more than I can handle. I'd never quite figured out the keyboard driver in SS. Your information about the @CTL function 255 may give me some clues.

Fm MISOSYS, Inc: Well, It's probably nothing that we would really want to spend time on. If you get some time to chat on the phone, I can discuss it with you. Essentially, the first thing that a keyboard driver does is scan the keyboard matrix (I'm assuming you have some familiarity with that technique). The result is an array of values. By scanning each row and then reading each column, the array of values represents a binary bit map of what keys are currently depressed in each row of the matrix. The rest of the keyboard driver then attempts to develop the ASCII value key-code based on the scan codes. SS uses the CTL-255 function which returns the array of values. A filter put between the DOS and the installed PRO-WAM, could possibly be written to take PRO-WAM's exported values and return them as the scan array for the CTL-255 function. It's a lot of work and would require some heavy thought, but I think it could be done. Sort of a "backdoor" approach to SS.

Gary Phillips To MISOSYS, Inc: Is that what CTL-255 does? I've been wondering about it on and off for two years now. Does Superscripsit really use it? What else uses it? (Guess I could have looked it up in the Source, but it hadn't crossed my mind since the books arrived.)

Fm MISOSYS, Inc: I hope that no one else uses it. Believe me, I really hated to have to put that into the DOS, but the alternative was worse.

Shane Dawalt To MISOSYS, Inc: I don't think RS had any choice. The Model III version of SS used the <@><whatever> key sequence for the control characters. The model 4 has a built in <CTRL> key. If RS would have broken "tradition" that could have confused SS users, e.g. use the <@> on the Model III but use the <CTRL> on the Model 4. This may be a little weak, but SS can keep track of the CAPS key internally without referring to DOS. I know I know, VERY WEAK.

Fm MISOSYS, Inc: That was their reason; to maintain consistency across the Model III and 4 for SS users. But look at the kinds of problems that level of incompatibility with the DOS has caused.

Fm MISOSYS, Inc: After much dialog between Lynn and myself, and much work by Lynn, he made great headway into the implementation of the CTL255 keyboard filter. I also made a little change to PRO-WAM so that it established a flag indicating that EXPORT was active. This flag, agreed upon by Logical Systems, is bit 4 of KFLAG$. The CTL255 filter is currently in one of the DLs of our forum on Compuserve. I have asked Lynn to consider submitting the filter for publication in THE MISOSYS QUARTERLY. I'll keep you posted.

Applications which don't close files

Lynn Sherman To MISOSYS, Inc: I just noticed a possible problem with the latest version of PRO-WAM. It may be a problem with the application I'm using, but it used to work fine under the old version of PRO-NTO. I tried to use a printer setup application called PRINT/APP that I downloaded from one of the DL's here a long time ago. It worked fine the first time I used it but when I tried to reenter it a second time I got a message that the PRCTRL/DAT file was not found. A check of the directory showed that the file was still there, but was left open. This may be a fault of the home brew application but it never showed up before.

I have not had a chance to examine the application program in detail. I just wondered if any changes were made that might affect the closing of an open data file on exiting PRO-WAM.

Since other applications that access data files still work fine I suspect it is a problem with the application. I just can't help wondering why it used to work, but doesn't now.

Fm MISOSYS, Inc: It may have never showed up before because that is controlled by the users's having the NETWORK flag set (most folks didn't). PRO-WAM 2.x automatically sets the network flag. The result is that the DOS's file open bit is in use. If a file is not closed, then it is "left open" to the directory. A subsequent open will get a "File already open" error and force to READ-ONLY. Unless the application checks for that condition, it's probably going to abort. PRCTRL/APP will most likely need a little revision.

Lynn Sherman To MISOSYS, Inc: I see. I took a quick look at the PRCTRL/APP and I saw that it will abort with the "DAT file not found" message if any error occurs in opening the file, which of course includes an open file error. I made a quick, but unacceptable fix by bypassing the error check. This allows the APP to access an all ready open file, but will hang the computer if }i if the DAT file is not present. I don't have the source file but will look into a means of closing the file before exiting.

Fm MISOSYS, Inc: The other alternative is to use the FORCE READ flag when opening the file. Set bit-0 of SFLAG$ just prior to opening the file and it will be set to READ ONLY - the "file already open" bit won't be set and the file will not have to be closed. Of course, that technique will only work if you don't have to write to the file.

Lynn Sherman To MISOSYS, Inc: Thanks for the info about setting bit-0 of SFLAG$ on opening the file. As it turns out, I have already fixed the PRCTL/APP to close the /DAT file on exit. Since the APP has an editing feature, I would have to have done it anyway I guess, or I couldn't change the codes from the APP.

PRO-WAM and ALLWRITE: Please read README

Fm Garry DeVol: I recently purchased PRO-WAM from MISOSYS. Since Allwrite uses the DOS keyboard driver, I thought that it would work, but invoking PRO-WAM freezes the computer. I also have Superscripsit 1.2.0, which works fine since I installed the SS12PW/JCL patches. I would like to be able to use Allwrite and PRO-WAM together as well. Another application which crashes with PRO-WAM is Profile 4+. Any information about patches for Allwrite or Profile 4+ would be greatly appreciated.

Fm MISOSYS, Inc: When all else fails, READ YOUR DOCUMENTATION! There is a note in the README/TXT file which advises you to apply a patch to your DOS because of a stack management problem. The patch is even on the PRO-WAM disk as a JCL file called FIXBOOT. Please read the README file. Also, very little works with PROFILE because that stupid package DOES NOT HONOR THE HIGH$ POINTER OF THE DOS!!! Invoke one of the modules of PROFILE and you will clobber what's up in high memory below about X'F800'.

LDOS Support To Garry DeVol: Roy is entirely right about the programs mentioned. RE Profile 4+, the EFC9 (Inquire, Update, Add) is the particular offender. As I recall, at least one person has modified EFC9 to be more respecting of the machine environment, but offhand I've forgotten who and how you can obtain their fixes.

Garry DeVol To MISOSYS, Inc: Thank you for your prompt reply. I have tried the patch and it works as advertised. I thought I had read the file, either I was mistaken or I was not aware that a 6.2 patch to BOOT/SYS would work on 6.3.0, which I am now running. Sorry. Thanks for your help, though. I have received the Spring TMQ, in the meantime, and have read it through for the first time. Maybe one of

these days I'll have the time and/or knowledge to make a contribution.

You can't invoke applications from EXPORT!

Fm Dick Guerin To MISOSYS, Inc: Hello again. Into Pro-Wam, one day later. From a cold start - I load a screen using Doload/app. Press F3. Run TODO or Cardform (APPs that access the disks). Exit TODO. Press <BREAK> to exit the screen loaded from Doload. Computer freezes up. Same results with 6.2 or 6.3, running from floppies, or 6.2 from hard disk.

I noted the patch to Boot/sys was not needed if the AT memory board patches were installed on the system disk earlier. Took a while to check it out. Suggest you mention it in TMQ so those not having the Source and with the AT memory board can save a few minutes during installation. Thanks.

MISOSYS, Inc To Dick Guerin: Dick, Your bug report has been investigated. Seems that that particular bug has been in PROWAM (nee PRONTO) since the original release; it just wasn't noticed. The crash occurs when you invoke an application while you are in the middle of a block mark operation in either EXPORT or IMPORT. If you were a 1.0 user, you probably never tried to invoke an application after you had explicitly invoked EXPORT (via <CLEAR RIGHT ARROW>). What happened was that when you broke back out of the application, you wound up in the MARKing mode. You were still in control, though. When the export is finished, control passes back to the RET statement which follows the window CLOSE SVC. However, the library overlay region didn't get restored with the application which was exporting, it got restored with the application which was invoked from the export MARK mode. The ability to invoke an application from the MARK mode has to be inhibited since it would even permit recursion via another export or import; this must be prevented!. I have revised PRO-WAM to 2.01a and a new disk will be sent out.

Activating PRO-WAM from JCL's //KEYIN

Edgar R Rondeau To MISOSYS, Inc: Roy, received Prowam release 2 and am trying to use it (learn it??) Can one execute prowam from a jcl, return to the jcl, execute (ctrl p) use sF3 to get command, do a directory search, <break> return to jcl and complete a keyin?? I tried it and the computer froze up. How does one access or create a dialer/app? or any other application? Btw, it works well with cobol programs. I've been able to ctrl p and use it to check on whatever. I think prowam is only limited by the imagination of the user.

Fm MISOSYS, Inc: PRO-WAM can be loaded form JCL. You can invoke an application using the PRUN utility. The entire capabilities of JCL operation are covered on page 12 under the section, "Automating PRO-WAM Installation".

Fm Edgar R Rondeau To MISOSYS, Inc: Roy, the problem was not in loading from a jcl, but returning to the JCl after having invoked the F3 key and looking at a directory and then returning to the JCL to execute the next program. It returned to the JCL menu with the cursor flashing at Keyin, but it would not accept the key.

Fm MISOSYS, Inc: Have you applied the BOOTFIX patch to 6.3? Some versions need it. That may be your problem. Other than that, I would have to see your JCL procedure. Didn't you get a software complaint form with the 2.00b release which asked you to explain the problem in detail? Depending on what you were trying to do, it may not even be possible.

Fm Edgar R Rondeau To MISOSYS, Inc: Roy, I did install the Bootfix. I run all my programs from a jcl menu One of the selections is Pro-Wam; others are cobol programs, basic etc. Each program, after completing, returns to the jcl menu from which next selection is made. The problem I encountered was when I wanted to back up a disk and used Pro-wam to check the directory on the disk. After using the break key, the program did return to the Jcl menu and flashed waiting for the keyin, but nothing happened. I had to reboot to get going again.

Fm MISOSYS, Inc: I still don't know how you invoked PRO-WAM from a JCL? Could you please provide me with in-depth details?

Fm Edgar R Rondeau To MISOSYS, Inc: Roy, hope I can explain it well. When I turn on the system a jcl menu is displayed from which I can keyin my choice. One of these choices is Prowam. After prowam is installed the jcl menu reappears and another selection can be made. If I want to invoke prowam at that point I respond to the keyin with ctrl P which gets me the window. Lets say, I use sF3. This prompts me for a command. I use dir :6. This displays the dir on drive 6. After checking my file I then hit break which returns me to the window

at which time I hit break and return to the jcl menu. So far so good. The cursor then flashes at keyin on jcl menu, but it will not accept the keystroke. So, I have to reboot. On the mainframe we say the keystroke was lost. Hope this makes it somewhat understandable.

Fm MISOSYS, Inc: I think I now grasp what you are doing. Receiving a copy of the JCL file which you use would have been preferable. However, it appears that you are using the //KEYIN facility of JCL to activate PROWAM. You then are using the LIBEXEC facility from PROWAM to issue a DOS call. Here's why that won't work. When you use //KEYIN, SYS11/SYS is the DOS overlay present in the OS overlay area. Once you activate PROWAM at that point, any number of other DOS overlays are loaded into that region. When you have completed your PROWAM work, PROWAM loads SYS1/SYS automatically. That's done to correct for the KEY input dealt with by SYS1's command interpreter. However, the program counter returned to is a location of SYS11, not SYSi. That's why your system is locking up. This operation should be the same as was done under PROWAM release 1. Did you have that release? If so, did you ever use that JCL procedure? I am not sure that I would be able to deal with that kind of operation. I'll give it some thought.

Fm Edgar R Rondeau To MISOSYS, Inc: If I activate Prowam from Jcl, then execute a cobol program and while in the cobol program I use ctrl c then I can use all of Prowam and it will always return to the cobol screen. When I end the cobol program, it will return to the jcl without any problem. However, if I am at the jcl menu and use ctrl-p, I can use the prowam programs, but when I break out of the Prowam I will return to the jcl menu, but the system freezes. This still makes prowam very useful.

Fm MISOSYS, Inc To Edgar R. Rondeau: I know I previously said that it would be some time before I resolved your PRO-WAM activation from JCL's //KEYIN macro; however, I really got curious about it. Since my last message, I delved into it further. I had to build a non-system debugger so I could step through SYS11 (DEBUG shares the same address space as all other system overlays and thus cannot be used to debug SYS11). With that in hand, I narrowed the problem. I patched PROWAM to re-load the system overlay which was resident when PROWAM was activated; it previously always re-loaded SYS1 which was the requirement. That patch behaved properly, and SYS11 was being restored when PROWAM exited from a //KEYIN activation. But here's the clincher. After SYS11 grabs the keystroke you entered to match up with the execution conditional block, it has to read through the JCL file looking for the proper "//?" statement (where "?" is the character 0-9). Unfortunately for us, it expected the JCL file buffer to be correct, i.e. to contain the current JCL sector which was being read. Too bad, though, because the system buffer is used for that. Certainly by now, that buffer contained something else. The crash was caused by SYS11 reading JCL line input until it found a carriage return. Normally, SYS11 will read at most 79 characters. However in the //KEYIN routine, it has no upper limit check. The JCL input buffer is the normal DOS input buffer which precedes the Drive Control Table (DCT). When that line buffer overflowed due to reading some non-JCL sector of data, it crunched the DCT. That's why the crash. You can crash it without the PRO-WAM activation by invoking a non-compiled JCL file with a //KEYIN macro followed by a string of characters which exceed 80 characters before a CR. Well to make a long story longer, I patched to PROWAM which now correctly restores the JCL file buffer if SYS11 was the overlay resident when PROWAM was activated. It works like a champ.

Now here are some cautions. (1) If you invoke anything from PROWAM activated from JCL which will use the @KEYIN service function of DOS (SVC-9), that input will be satisfied from the JCL file - not necessarily what was intended. Also, you should not invoke a DO command from the "Command?" LIBEXEC mode of PROWAM because there is a JCL file already in effect - unless that is the intent (I don't know, offhand, what the end result would be). The patch is WAM21/FIX located in The Patch Corner. Have fun and enjoy.

TODO bug: TODO fixed!

Fm Paul Rehberg To MISOSYS, Inc: TODO bug: If you call up a TODO file containing more than 1 screen of entries, and use the ADD function on the first screen, TODO does not place the new entry at the end of the datafile; instead, it puts it as the first entry on screen 2 (entry #9) and wipes out the remaining entries.

HEAD query: Why does HEAD omit the company name if blank (a good feature), but not do the same should the last name, first name field be blank? I have many such entries to companies that do not have an individual's name; the

blank between the company name and street should be eliminated.

HEAD wish: My preference is to put the individual's name first, followed by the company name on line 2. Is a quick patch to reverse the two pieces of data feasible? I'd rather not open the can of worms on which way is the "correct" way.

PROWAM manual: a good piece of work. Rather than mentioning WORD in the next quarterly, you ought to be praising ALLWRITE (better late than never). That program and PRO-WAM are the only things keeping me on a 4. Style sheets can never make up for WORD's sluggishness on all but the fastest machines. I'll bet ALLWRITE's editor on a 4 is faster than WORD on your AST, right?

Fm MISOSYS, Inc: I'll check out TODO. HEAD operates that way because that's how its programmed. A simple patch would not be able to change it. If you know assembly, and have access to PRO-DUCE, it wouldn't take much to re-program it. But customization of that nature is not something to put into an off-the-shelf facility such as PRO-WAM.

Incidentally, the PRO-WAM manual was done using Microsoft WORD, not ALLWRITE. We used ALLWRITE on TMQ Volume I; it did a commendable job. ALLWRITE's editor is faster on a 4 than WORD on a PC (not necessarily my AST). But I do get to see all my formatting on-screen as I prepare my text with WORD.

Programmer's Toolkit? When?

Fm Dick Guerin To MISOSYS, Inc: [PRO-WAM;] Nice program. How much for the Programmers' Toolkit? My Winlink don't work no more.

Fm MISOSYS, Inc: We haven't even begun to think of a price or a package for the toolkit. On the other hand, I wouldn't have expected the older WINLINK to not work any more. True, I never tested it on the new version; however, the SVC interface changed very slightly. What problem are you having?

BRINGUP, WINLINK, and PSORT

Fm Paul Rehberg: When can we expect WINLINK 2.0? I've had to revert to PRO-WAM 1.0 since WINLINK won't work with 2.0, and I have two programs that I use regularly that require it.

Here's two more bugs for you in 2.0: BRINGUP: Under the ADD function, <BREAK> does not abort when entered as a response to the Text? prompt. Instead, you are re-prompted for Priority and Time, then the text field is filled with garbage from a previous keyboard entry. I was surprised to see my previous ADDRESS entry appear in my BRINGUP entry.

PSORT: The pack parameter causes an error if you use the closing ")". Thus "(P" or "(PACK" will work, but not "(P)" or "(PACK)". Also, if you let PSORT prompt you for the filename, it will abort unless you include the /DAT extension; PSORT somehow incorrectly adds the extension, resulting in a file not found message.

Fm MISOSYS, Inc: Paul, I think you will find the following patch will correct WINLINK:

```
PATCH WINLINK
 (D01,C6=57 41 4D 03:F01,C6=50 52 4F 4E)
```

Winlink would be doing a @GTMOD on "PRONTO", the module header name of release 1.0. The module name of release 2.x is "WAM"!

I worked up a patch to correct BRINGUP's correctly aborting the "Text?" input to the ADD command on <BREAK>. It is BRINGUP1/FIX and should be applied to both WAM1/APL and WAM0/APL. The patch is in The Patch Corner.

Here's the reason for the bug: BRINGUP under 1.0 actually had an un-reported bug I discovered when doing release 2. The bug occurred whenever you tried to enter a text string which was exactly 29 characters. BRINGUP used the actual file buffer location for the text string. But the @WKEYIN function terminates the input with a CR. We knew this and subsequently stuffed the CR position with a SPACE. But if the full 29 characters were input, the CR and subsequent SPACE were written to the first position of the next record altering it if it was an active record. Funny how no one reported that. Well to fix up that bug, I recoded BRINGUP to use the ECHO text buffer storage for input of the text line. However, the first position of that buffer is also used to indicate that the buffer is active when it is non-zero. I completely forgot to consider the result if you hit <BREAK> in response to the Text? prompt. Since the first byte would be an 80H (or some other text if you entered a partial line then hit <BREAK>), when the ADD

aborted, the main command input routine would sense that a pending ADD was in effect and would automatically go into the ADD mode again! That's how ECHO and MOVE operate. You only got prompted for the Priority and Time because ADD now used the ECHO buffer. You must have previously used ADDRESS; the address record fragment remained in the ECHO buffer when you broke from BRINGUP's text add.

Finally, I don't see any parentheses on page 25 of the PRO-WAM manual. It is true that PSORT will ignore a left parenthesis prefixed to the "Pack" parameter; however, there is no mention of it in the documentation. Thus, if you follow the documentation, it works as advertised. As an aside, The parameter service call of the DOS does not require a closing parenthesis. PSORT, by the way, does not use the DOS @PARAM service call; PSORT is written in C.

On the other hand, I do have a solution for the default file extension problem. I never tested it being prompted for the data file specification since I always type that on the command line. I suspect that all of my beta testers do the same. The problem stems from the differences between the gets() and fgets() functions in C. I thought I was using fgets() which strips the newline character from the line input. Since gets(), which I was using, keeps the newline, the default "/DAT" extension was appended to the filename string after the newline character. That, of course, was not too useful. PSORT22A/FIX in The Patch Corner fixes that up.

Minor CAL bug: CAL fixed!

Lynn Sherman To MISOSYS, Inc: Roy, I have noticed what I think is a minor bug in the CAL application. If you page forward through the months using the "N" function and move to the next year, the date displayed at the upper right changes to January. If you page backward it changes to December. The year changes also. It seems to be connected to the updating of the screen that occurs at the year change. I thought this was to remain constant as Today's date and only the one on the left should change. Sorry I didn't notice this sooner.

Just some additional info on that CAL/APP problem I mentioned. The "Today's Date" is not modified on year changes if there is no BRINGUP/DAT file present. Its apparently happening in the routine that checks for BRINGUP activities. The date change occurs both when "paging" using N or P and if you move from one year to the next with the arrow keys (as from Dec 31 to Jan 1).

Fm MISOSYS, Inc: Yes, the current date in CAL changes when you transfer from one year to another IF you have a BRINGUP/DAT file. I don't think I can do much about that. I'll look into it, but space in CAL is very tight; I think I have one free byte. The large apparent free space is used for the day flags (there's 366 of them) reflecting BRINGUP activities.

Well after perusing the issue, I found a way to correct the problem. That was due to the entire screen refreshing after before displaying a date in a year different from the current. My solution was to poke a RET instruction into the routine which refreshed "today's" date, since that was the last part of the screen update. In order to gain patch space, I had to re-sequence the command line display putting "Bringup" last. In that way, I was able to reuse that command string to invoked the BRINGUP application via the "B" key. I then shortened the message preceding the original "bringup" execution string by one byte to gain the nine bytes needed for the patch. That should be CAL1/FIX in The Patch Corner.

Sorting PRO-WAM data

Automated Sorting/Searching PRO-WAM Data Files

By Doug Tittle

MISOSYS has recently begun shipping Release 2 of PRO-WAM. The new release contains many useful new features. Among these are the addition of a second sort field and the linking of the Address Book Search feature to the first sort field defined in the data file header.

These new features make it easy to use procedures (/JCL Files) which can make applications like Address and Card(X) behave more like data bases. The general idea is to write a JCL procedure which allows you to change the sort fields to whatever you wish and then sort and/or search on these when the application is invoked. Since you merely have to run the procedure to change the order of the data, the mild mannered Address Book and Card/CardX applications become powerful mini-database managers!

It's easy to create these JCL files. You'll need the PRO-WAM documentation and a text editor. Folks with the Mr. ED package from MISOSYS can even do the entire thing from within PRO-WAM itself! Let's go through making a procedure for use with the Address Book. The finished JCL file is included on this month's Disk Notes for your enjoyment.

First, look at your PRO-WAM documentation and note that there are two areas of three bytes in the header of all data files which specify the location and length of the two sort (key) fields. They are in the first sector of the data file at 14H-16H and 1AH-1CH (all numbers are in HEXadecimal). Our procedure will use the DOS PATCH utility to change the contents of these areas so that the file can be sorted and searched on ANY field.

Next, look at the Technical Information for the application you want to work with. In our case, this means looking on page 45, where you will see the following chart: (I've added the HEX translations)

Field Ident	Field Label	Width	Record Location
0	Last	15 (0FH)	000-014 (00H-0EH)
1	First	10 (0AH)	015-024 (0FH-18H)
2	Company	20 (14H)	025-044 (19H-2CH)
3	Address1	20 (14H)	045-064 (2DH-40H)
4	Address2	10 (0AH)	065-074 (41H-4AH)
5	City	15 (0FH)	075-089 (4BH-59H)
6	State	8 (08H)	090-097 (5AH-61H)
7	ZIP	10 (0AH)	098-107 (62H-6BH)
8	Data1	5 (05H)	108-112 (6CH-70H)
9	Data2	12 (0CH)	113-124 (71H-7CH)
10	Flags	3 (03H)	125-127 (7DH-7FH)

The chart gives us all the information we need to set up the data file's header to produce sorts on whichever field we want. Our next step is to write PATCH commands for each field. Here's an example for the City Field:

```
PATCH Address/Dat (O=N,D00,14=4B 00 0F)
```

The O=N parameter (that's the letter OH, not Zero) prevents PATCH from looking for the 'F' statement. It allows us to alter the field without knowing what the old sort field was. The patch location is for the first sort key. The last three HEX numbers came from the chart. Note that the first two define the location of the sort field in the file and are reversed. That is, the City field is really at '004B', but is entered as 4B 00. This is a standard Z80 convention. The last number is the key length, and is always just 1 byte. All 10 PATCH lines are shown below:

```
Patch Address/Dat (O=N:D00,14=00 00 0F)   Last
Patch Address/Dat (O=N:D00,14=0F 00 0A)   Frst
Patch Address/Dat (O=N:D00,14=19 00 14)   Comp
Patch Address/Dat (O=N:D00,14=2D 00 14)   Add1
Patch Address/Dat (O=N:D00,14=41 00 0A)   Add2
Patch Address/Dat (O=N:D00,14=4B 00 0F)   City
Patch Address/Dat (O=N:D00,14=5A 00 08)   St.
Patch Address/Dat (O=N:D00,14=62 00 0A)   Zip
Patch Address/Dat (O=N,D00,14=6C 00 05)   Dat1
Patch Address/Dat (O=N,D00,14=71 00 0C)   Dat2
```

Now that we have the patch lines, we're almost done. The patch lines for the second sort field can be derived from those above by changing the PATCH location from D00,14 to D00,1A.

All that remains is to add some selection ability using JCL comments and the //KEYIN function. I find that a useful method is to select the first field in one procedure which patches the first key location, then passes the selected field name to a second procedure which handles the second key. The second procedure then calls PSORT if desired to sort the data.

Here is a fragment of such a JCL file:

```
.Address Book First Key Field Selection
.
.Press <1> for Last Name  or  <2> for Company
.
//KEYIN
//1
//Assign K="Last Name"
Patch Address/Dat (O=N:D00,14=00 00 0F)
Do SortAdd (@Part2,K="#K#")
//2
//Assign K="Company"
Patch Address/Dat (O=N:D00,14=19 00 14)
Do SortAdd (@Part2,K="#K#"
///
//Exit
@Part2
.
.Address Book Second Key Field Selection
.First Key Field is #K#
.
.Press <1> for Last Name  or  <2> for Company
.
//KEYIN
//1
Patch Address/Dat (O=N:D00,1A=00 00 0F)
PSort Address/Dat
//Exit
//2
Patch Address/Dat (O=N:D00,1A=19 00 14)
```

```
PSort Address/Dat
//Exit
///
//Exit
```

You can add the PACK parameter to the PSort command lines if you wish. Be careful, however, since any field which begins with an FF hex (255 decimal) character will be deleted! I use PACK when I'm sorting on the default key since this is where the FF is placed when you request a delete in PRO-WAM.

This technique of changing the sort fields works with ANY PRO-WAM data file, although I've shown Address. You'll find that once JCL files are created, manipulating the sort and search features is useful and convenient.

With a little practice, it becomes easy to create these procedures. Soon you'll wonder how you ever got along without them!

[Editor's note: The entire JCL file has been included in DISK NOTES 2.1.]

RATFOR

RATFOR Testimonial

Fm Gary Phillips: Unsolicited recommendation: I just received my copy of RATFOR-M4 from MISOSYS and I am very favorably impressed. Jim Beard and Roy Soltoff have done an excellent job in pulling together the manual and the software, and with all the stuff you get, it's well worth the $99. If you are interested in structured programming, or FORTRAN, or language implementations that are portable across many different kinds of hardware, you owe it to yourself to take a serious look at this package. I find it will allow me to write a single, nicely structured program that can be compiled and executed on model 3, model 4, IBM-PC, IBM 370 mainframe, VAX and MicroVAX! That's not an insignificant capability...

RATFOR's QRAT/JCL and other matters

Gary Phillips To Jim Beard: Just received my RATFOR from MISOSYS and I am very impressed with the job you did pulling it all together, Jim. I had been considering doing it myself from the K&P source code, but with the sample code, documentation, and LED included, the MISOSYS package was easily worth it.

Now for my question: using the F=filename option of QRAT. If I specify a file that doesn't already exist, the JCL aborts on the REMOVE #F# statement. If I pre-CREATE an empty file, and specify that on the DO command, it goes only as far as creating its own copy of the file, after having removed mine. Then it finishes with ".Job done" and no compile. Now if I run QRAT another time, all works as expected. What am I doing wrong? I admit I'm not fluent in LDOS/LS-DOS compiled JCL, but there do seem to be some odd things in QRAT and I don't see any details in the manual beyond how to invoke the file.

Fm Jim Beard To Gary Phillips: I gather from your question that you have the Model 4 version. There is a serious problem with using JCL with LINK *DO to *XX (where *XX is a dummy devspec set to a filespec). If the file does not already exist, the JCL will crash. My patch was to use CREATE, which still has problems. There is no perfect solution at present. The problem is with the way that JCL works with DOS. This can be very tricky when *DO is involved. [editor's note: See the discussion on JCL in LS-DOS Information for the answer to this problem] What I would suggest is that you COPY any convenient file to the filespec you want to use so that the JCL doesn't crash on the REMOVE.

I know that this feature is nice, but you should only need it when your printer can't reproduce the graphics characters that are used to walkback macros when nested macros are result in errors. Try the hard copy option and use the printer; this is more efficient for debugging anyway. If you have any more trouble with the F= option, let me know.

Gary Phillips To Jim Beard: Certainly not a critical problem. As I said, the package is pretty impressive. I am going to have to study QRAT further because I have difficulty understanding why you are doing what you did in there anyway. Perhaps I'll find a solution; usually my way around glitches like the one you describe would be to write a small utility program to handle the job, and invoke it from JCL. I'll get back to you after I analyze the situation further.

Jim Beard To Gary Phillips: The MISOSYS RATFOR is far more general than that you could get from "Software Tools" alone. Your RATFOR has all the keywords of the UNIX version, which

are not all discussed in the book, much less given as skeleton code. The macro expansion discussed there is the basis of MS-DOS JCL argument expansion among many other software packages and modules used ubiquitously, but the book doesn't integrate it with the RATFOR interpreter. In addition, you have multi-level "break" and "next", which means that you will NEVER be forced to a "goto". There was a journal paper in ACM about this while the C language and Berkeley UNIX were being formulated, and this feature is seen only in MISOSYS and UNIX 4.2 RATFOR.

The applications files were intended to provide several examples of SERIOUS software using RATFOR. In addition, they are sufficient to get you started in several different areas. Some will prove valuable to those who don't even use the language, such as the filter design examples.

In RATFOR or any other language, your own programs are your property. I claim no copyright on FORTRAN files produced from source code written by anyone else in RATFOR. My copyright is only on the translator and other utilities on the distribution diskette.

Microsoft, the copyright holder for F80, has a policy in effect at this time on all their languages. This policy is that programs linked to their libraries must carry a copyright notice identical to that of the compiler copyright message which reserves the rights to the Microsoft run-time modules. If they didn't do this, the /CMD files could be disassembled and their library could be used without a Microsoft copyright.

Since F80 is the only widely distributed FORTRAN for the TRS-80, we felt that we could avoid this issue by distribution of a /REL file. For MS-DOS, this is not true, since Ryan-McFarland also sells an excellent FORTRAN which has been fairly widely distributed for some time, and we distribute an .EXE file with copyright messages as required by Microsoft.

Phil Oliver To Jim Beard: I understand you're the author of RATFOR, so perhaps you could elaborate a little on the enhancements RATFOR provides to usual FORTRAN. I take it RATFOR preprocesses RATFOR code and produces FORTRAN output?

Jim Beard To Phil Oliver: An introductory blurb on RATFOR is in DL1 as RATFOR.DOC. As an engineer (FORTRAN user), what RATFOR gives is the ability to forget column 7, statement continuations, DO loop terminations, special lines for comments, the invisible hungry demon at column 72 that invisibly eats the rest of your statement, and most other things that you have to worry about when you use FORTRAN. After 20 minutes to a day (depending on your receptiveness to the different notation) you can do the algorithms without concerning yourself with the language any more.

New users with no FORTRAN experience will find RATFOR as forgiving as BASIC on the statement level with the ability to encapsulate subprograms completely. RATFOR subroutines and functions are as separate as BASIC keywords. Writing a set of subroutines and functions is much like generating your own language.

RATFOR software concepts aren't new; they predate C and were invented by the same people that invented C. RATFOR looks a lot like C except for the declarative statements, which are shared with FORTRAN.

Language development for RATFOR continued for a while after C was frozen. The result was a couple of generalizations, including an easier to use "switch"/"case"/" default" branching construct, and a multi-level "break" and "next" capability that C does not have. The latter means that a good RATFOR program will never be forced to a "goto", something that cannot be said for even C.

They use police radar guns to time baseball pitches almost universally. To be accurate, they have to be behind the plate. This means that sometimes the umpire or catcher or batter will block the view, and that you have to be in a crunch of fans, limiting mobility. A smart radar gun is one which can be off the line of flight. The example in the RATFOR package is a Kalman filter which will give accurate velocities from any angle. I show how to use the TRS-80 to develop the software and hardware prototype using RATFOR. Included in the example is a complete model of the situation with screen graphics outputs, which you can play with to get a feel for how the radar gun will work.

The Hardware Corner

64K -> 128K on 4P

Fm LDOS Support: No PAL needed for the 4P. The instructions will fit in three sentences.

(1) Open machine and find the eight empty sockets on the motherboard. (2) Plug in the chips (pin 1 as indicated), and move the jumper (E1-E2-E3 or E11-E12-E13). (3) Put it back together and test.

Tandon TM100-2 Drives

Fm LDOS Support: The drives should have two shunt bars each in place. The proper drive select for each drive (DS0 for the first and DS1 for the second, assuming that the shunt positions are numbered from DS0 to DS3) and the HM bar. MUX, HS and the remaining DSns should be open. Lastly, you need a drive cable that has all pins intact, not missing pins as the RS cable is. If you are going to mix one of these drives with a stock RS drive then you need to do things a bit differently.

XLR8 Discussion

Fm Jim Gaffney: To use the XLR8er with both HD and RAMDISK requires a little trickery. Unfortunately, all the drivers for the XLR8er as currently implemented reside in low memory. If you already have your HD drivers there, you run into conflicts - as you have already discovered. What you need to do is to rewrite a copy of your HD initialization JCL to: (1) Install the FIXALL filter if your system requires it. (2) Install a RAMDISK - any size, as long as the driver is loaded in lomem. (3) Install your HD device drivers - which will now be forced into himem. Do NOT reformat - only install the drivers, i.e., SYSTEM (DRIVE=4,DISABLE,DRIVER="WHATEVER") and the subsequent questions. (4) Install any other device drivers that you want sysgened. (5) REMOVE the RAMDISK driver - i.e. (SYSTEM,DRIVE=n,DRIVER="RAMDISK") and the subsequent answer to disable. You have already created space for the driver when created later. If you don't have 8 other logical drives, you can leave it in place. (6) Sysgen the whole kit and kaboodle. I'm using a Percom drive with Powersoft drivers, but I'm assuming that the RS drivers will relocate into himem if there is no more space left in lomem.

Besides the Model 4 drivers, there are also LDOS drivers written to give a 250+ RAMDISK on the Model 4 in the Model III mode - as well as drivers for MM's CP/M. I got mine from the software authors, but H.I.Tech ought to be able to furnish them to you if interested.

Fm Martin L. Beauchamp: I've been noticing some questions and comments about the XLR8er board on the forum lately, and I thought I'd throw my two cents in.

I'm running an old Model 4, 128K, white-screen, non-gate array with three RS 5-meg HD's and three 40-trk, SS floppies. Some of the newer MISOSYS offerings prompted me to look for more RAM and some speedup of the CPU.

The wife bought me the XLR8 for Christmas, and after a comedy of errors (including some blown RAM due to sheer stupidity) it's up and running.

Using the 256K on-board additional memory for a RAM system drive, speed improvements in /CMD files of 200-300 percent have been noted. The greatest advantage is seen in programs which access disk files frequently. These improvements are over and above using the MemDisk as the system disk under the Z80 configuration.

So, the Hitachi HD64180 does do what they say. A recent mailing from H. I. Tech shows: User Manual $10.00 (refundable on purchase of the board, but definitely not up to the standards

we've gotten used to thanks to Roy and his kind). RS-232 Option $69.95 (provides 2 ports); Ciarcia Buss Option $10.00; Additional Operating System Disks (CP/M, TRSDOS 6.2, LDOS 5.1) are available at $15.00 each. Note, these contain only the utilities, not the OS. Hitachi HD64180 manual $10.00

One point to be made to those considering this board... In order to realize the full speed-up of the HD64180, the RAM must be 150ns or faster. Slower RAM will fall apart at the less frequent refresh rates allowable by the Hitachi chip.

I've also noticed that, while LDOS 5.1.4, TRSDOS 6.2.1, and NEWDOS 80/V2 play well with the board installed, TRSDOS 1.3 is flakey. I'm going to run some tests to find out what the problem may be, and report back later. I know, I know, I should upgrade to LS-DOS 6.3 and LDOS 5.3. That's my next order of business once I've got this new board fully settled in.

Also, with this monster, I'm running a RAM system disk, PRONTO/PROWAM, and still have room for five text buffers under Pro-SAID. Now that's class!

Fm Dick Newman: I recently installed an XLR8er board and am using "TRSHD6" as my harddisk driver. I modified ALPHA1.FIX (DL6) to work with LS-DOS 6.3 and it allows me to remove FIXBANK. This frees up 246 bytes in low memory. The patches to Roy's ALPHA1.FIX to make it work on LS-DOS 6.3 are as follows: BOOTAT/FIX and SYS1AT/FIX are OK as is. SYS0AT/FIX needs these changes: Change D0C,CD= to D0C,D6=. Change D0C,DD= to D0C,E6=. Change D0C,ED= to D0C,F6=. Change CD 47 21 to CD 94 21 on this same line. Change D0F,E1= to D10,2E=. Change D0F,F1= to D10,3E=. Installing Roy's patches with a harddisk as system drive is very tricky. BOOT/SYS, SYS0/SYS, and SYS1/SYS (all patched) must be on the floppy. SYS1/SYS (patched) must be on the harddisk (if you plan to make the harddisk the system drive). The 256K Ramdrive also makes a great system disk. I found I can load the Ramdrive with FDR6AT (the new debugged one, found in NEWRD.ACH) and then install the RAMDISK driver without formatting. The program LOMEM was very helpful as was MAPMEM by Donald Brandt.

Fm MISOSYS, Inc: The revised alpha fixes for 6.3 were in FIXES6. One way I use my boot disk under 6.3 is to SYSRES a patched copy of SYS1. Then it becomes part of the configuration. Since SYS0 and BOOT are taken off of the floppy, the hard drive files remain "stock". Then it is easy to boot a non-patched copy of 6.3 if I want to exclude the alpha board from a particular work operation.

Fm Gary Phillips: Anyone else using XLR8er on a 4P run into this one yet? Despite the (weak) documentation stating otherwise, my machine can no longer cold boot anything requiring MODELA/III. It fails to find the ROM image file on the disk, or else gets a LOST DATA error trying to load it. My conclusion is that the cold start configuration of the 64180 (3 memory wait states, 4 i/o wait states) is actually too slow to keep up with the FDC for an extended load like the ROM image requires. Unfortunately, this configuration is recreated every time you press the Reset key.

Short of burning a new boot EPROM (may get around to that) to change the CPU speed before attempting to boot, my only possible work-around is to boot LS-DOS first, use SET180 to appropriately set the CPU speed (SET180 (m=1,i=2)) and then use the BOOT library command to start up model 3 mode. Any comments or similar experiences?

Fm MISOSYS, Inc: See Gary Phillips article on this very topic.

Fm David Hall: I got TMQ Vol 1 issue iii, and enjoyed it a lot. Just one small word of encouragement. I don't necessarily like your position concerning the future of support for Model(s) I/III/ and 4, but I understand completely what you are trying to say. I just wanted you to know that there is at least one customer out here who appreciates what you have been able to do in the past, and who understands that changing times sometimes require changes.

I see from the last MQ's that there seems to be a lot of interest in the XLR8er board. I have been using the HD64180 (and Zilog's second source) chip for about a year and a half now. As far as I am concerned, it is a SUPER chip. I have a Model 4 and 2 Model 4p's, all with 64180's in them. If you would be interested I would be happy to try and write something about the chip and the board for the MQ [editor's note: David's first article appears in this issue].

I have been reading your SIG with great interest what I have been able to find concerning the 64180 add-on. I thought it was about time for me to put my 2 cents worth.

Depending on which 64180 you are using, the MMU can indeed directly address 1 meg of ram! The DIP version however can only address 512K. The chip carrier versions can handle a full meg. Even with the 512K versions of the chip however, there is no reason you can not implement an external banking scheme to allow multiple banks of 256K rams.

True, the 64180 has two built in ASCI channels, and they do indeed have built in BRG's. However, there is a bug in the chip! DCD0 (DCD input on ASCI channel 0) is supposed (according to the documentation) to produce an interrupt on level change. In fact, the DCD0 interrupt is asserted when ever DCD0 is OFF. Hitachi has been unable to tell me how to get around this other than by tying the DCD0 input pin low. The problem is that not only does the receiver get held inoperative, but the interrupt cannot be cleared! It is possible to send "break" (long space) using either ASCI port by use of a little trickery! If the CKA leads are not needed for DMA purposes, and can be tied high, then by sending a byte of X'00' while the ASCI is configured for internal clock generation will start the spacing condition on the RS-232 line. By re-configuring the ASCI port to use external clock, it is possible to prolong the space condition (through the use of a software delay, maybe using one of the timer channels) for as long as needed.

[To clarify the above,] What I meant to say was that if interrupts for ASCI port 0 are enabled, then an interrupt will be asserted (with no known way to clear it) as long as DCD0 is "off." If interrupts for ASCI port 0 are disabled, there is no problem. My solution (however) has been to pull DCD0 to an "on" level through a p/u resistor to +5v, and then be able to use the ASCI port 0 interrupts for Tx and Rx at least.

Finally, I have seen stated somewhere, that the built in DMA controller will work with either ASCI channel. This is correct, but it is important to note that only DMA channel 0 can be used with the ASCI channels. As a matter of fact, the DMA controller works very nicely allowing DMA of blocks up to 64K.

More on Alpha Technology patches

Fm Duane Saylor: Here's a patch to fix TRSDOS 6.2 or LS-DOS 6.3 that has the Alpha Technology Bank patches installed. Fixes the access to Area 2 and Area 3 when using Allwrite. I have suspected that this patch was needed for several months but only had my suspicions confirmed last night when Roy Soltoff left a message to David Huelsmann.

```
.ALLWRITE/FIX by Duane M. Saylor 21-Mar-87
.Apply via PATCH BOOT/SYS.LSIDOS ALLWRITE
D06,37=70 23;F06,37=23 70
.EOP
```

Fm Ken Kane: Ah, sweet relief! Duane Saylor's fix to BOOTAT/FIX fixes my outstanding problem using the alternate editing AReas of ALLWRITE! I am using the NEWRD/DCT driver from NEWRD.ACH with Roy Soltoff's AT patches. I used (patched) BANKER from TMQ vol I issue 2 to control the bank which is used by PRO-WAM; and what used to be a house of cards is now a brick edifice. And no more frozen keyboards with PRO-WAM.

Special thanks to you (and jjkd et al) for guiding me through LIBrary and USQ operations so that I could get at the AT drivers. Now I can use the rest of the DL's more freely, too.

ALLWRITE can normally use Banks 1 and 2 for editing after the BOOTAT/FIX patch sequence is changed from 23 70 to 70 23.

Six weeks ago there were two deterrents to using the AT patches with the Alpha Tech expansion RAM: 1) BANKER didn't work, so I couldn't control the bank to be used by PRO-WAM, 2) ALLWRITE's AReas malfunctioned. Now both are fixed, and I am pleased to report that PRO-WAM is FIRMLY installed in Bank 3 with no conflict with ALLWRITE.

Thanks for support of the expansion RAMs. The Model 4 with hard disk and expansion RAM is one terrific system.

I used the Anitek RAM driver for Alpha Technology RAM for the first six weeks. Because it forces you NOT to use Roy Soltoff's Alpha Tech patches, you can't use BANKER to control the placement of PRO-WAM, or access the higher RAM for use with SAID or any other present or future MISOSYS products that use the AT expansion RAM.

The Anitek RAM driver doesn't do anything that you can't easily accomplish with an installation JCL. I just started using the driver from NEWRD.ACH in DL6, (you need ARCH4.CMD to extract the driver), with the AT patches AND Duane Saylor's patch to the BOOTAT/FIX patch; and the whole thing is working smoothly!

Fm Bill Schaper: The AT patch (from Roy Soltoff to allow upper (above 128k) bank access in error (as pointed out in TMQ I.iii page 110). I use the XL8er board and recently acquired a 5 meg hard disk. All was A-okay until I tried to use the upper 10 banks as a ramdisk. Alas, no Lo-Memory available. I remembered the AT patch that Roy offered in the Fall Edition. My Model IV (non-gate array) would lock up on boot. The patch line that is listed as D06,37=23 70 47 3A 78 00 30 02 E6 7F 77 E6 FC F6 82 ED should be D06,37=70 23 47 3A 78 00 30 02 E6 7F 77 E6 FC F6 82 ED. With the 70 and 23 reversed it works fine under LS-DOS 6.3.0. BTW this is the 3rd line in the BOOTAT/FIX patch.

Blooming CRT

Fm Ted Pinkert to Fred Oberding: I finally got around to following up on your message regarding the blooming CRT ("shrink problem on TCE video board #8790615"). As you may recall, I have this problem which was not cured by installing an Astec power supply and low-power Teac drives (obviously, I didn't do all those things just to try to solve the CRT problem). When I last talked to you, I said I would check out the part# on my video board and get back to you for specifics. Here I am, finally! I do have board #8790615. You mentioned Tech Bulletin VID:2 and its instructions to install a 12V regulator and an electrolytic capacitor. Do you have the details available? What specific parts are required? To what exact points on the board are they installed?

Fm Fred Oberding: First we need to determine whether the mod has already been installed. If you have a 6 inch or so yellow wire going from near the flyback transformer to the middle of the board and have an IC202 & a C212 near the middle of the board where the yellow ends, your board has a factory mod. If this is not the case you will need a 78L12 regulator and a 47 ufd/16 volt electrolytic capacitor.

(1) Lift the anode side of D201. (2) Lift the positive side of C211. (3) Connect the anode of D201 and the positive side of C211 together. (4) remove wire jumper J204. (5) Connect a wire between the connection made in step 3 & resistor R214. Use the hole vacated by J204 nearest R214. (6) Connect the positive lead of a 47 ufd elect. cap. to right side of R212. Connect the other end to ground. (7) Take a 78L12 and identify the pins. With the flat side towards you and curved side away, note the pins numbered 1,2,3 from left to right. Connect pin 3 to the positive side of C313 and pin 2 to the ground side of C313; this is on the solder side of the board. Use insulated sleeving where necessary. Add a six inch or so piece of #20 wire to pin 1 and attach it to the trace connecting the right side of R212, R207, R213 and pin 12 of IC201. If this isn't enough I guess I could mail you a copy of my copy of the mod. Also note that there is a new mod to all Aztec power supplies to replace C-9 (maybe different number on some P/S boards) a 220 ufd 10 volt elect cap with a 16 volt.

I had this problem after several months of flawless video with the video board mod.

Disk drive questions

Fm mark harris: Gotta couple of questions that I would appreciate some answers to:

(1) My 40tk SS drive has never worked very well. Even after alignment, it still gets parity errors occasionally. I was thinking about trashing it and replacing it with a 80tk DS drive. Since I'm replacing drive 1 in a model IV, I don't need a power supply or case. (Right?) $140 bucks or so for this kind of drive seems quite reasonable. Do 80tk DS drives have any special problems that I should know about and will what I want to do work?

(2) Radio Shack sells a hard disk controller #26-1138. Is this a necessary item for all hard drives or just some of them? I was under the impression that when someone bought a 3rd party drive that all he got was a cable, driver software, and the drive. What gives?

Fm Bob Haynes: All HD bubbles require a separate controller, but in the TRSDOS world we're used to them being packaged in the same housing. Most Model 3/4 third party drives are that way, so are the Tandy 5 and 15 Meg versions.

Because the 10 Meg HD (25-1025) is designed for use with either a 4 or 1000, different controllers must be used, and are therefore specified as separate accessory items. Hadn't you noticed how cheap (for Tandy) the 10 Meg was? No board, that's why! The board you mention (26-1138) is specifically for mating the 4D and Tandy 10 Meg systems.

Upgrading/replacing only drive 1? Sure, the P/S will handle it. But most replacements are 1/2 height drives nowadays, so you might consider filling the hole with (2) 40 Tk DS drives instead. You'll have to deal with the jumpers and changing the data cable. Unless you have at least (2) 40 trackers on-line, backups will be a bitch. Add the 80 trackers only for extra power, not as your primary drives. Do note that adding the extra drive internally can cause P/S problems, unless you get the low-power versions (like the TEACs).

If you're on a tight budget, I've a standard full size Tandon 40T/SS/DD in good shape you can have for $20 + $5 s/h. Will set up for drive 1 so you can just stuff it in.

Drive speed re-re-visited

Fm LDOS Support: The biggest reason for altering the speed of the drive from exactly 300 RPM is that the rotational rate of the drive is then 200 ms per rotation, an integral multiple of the interrupt rate of the CPU (25 Hz, 40 ms for the Model 3, 50 Hz/20 ms for the Model 4 mode).

Altering the speed from exactly 300 prevents the interrupts from synchronizing with a point on the floppy for more than a revolution or so. Interleave really doesn't enter into it, in the sense that altering the interleave would not remove the need to adjust the drives off of 300 RPM.

Once you accept the need to adjust the drives off of 300 RPM, you can go slower just as well as going faster to solve this problem. In TRSDOS 6.2.0 and later, and LDOS 5.3.0, the SYSTEM (Smooth) parameter can be used to eliminate the problem from the other end, by restricting interrupts more closely during disk I/O. That costs you type-ahead as well as real time clock accuracy (as though it were accurate to begin with, ha!).

Hard drive wanted

Fm LDOS Support: Ok, first off, you need a hard drive. As an example, the Seagate ST225 is not a particularly fast drive, but is very commonly used so is very inexpensive. For example, grabbing a handy IBM magazine, I see advertised all over the place for about $350 with an IBM controller. You can either ask to get it without the controller, which will probably save you about $50 to $75, or go ahead and get it with the controller so that you can pop it right into an IBM clone if you ever decide to get one.

Stay away from the ST238, or anything advertised as 30 meg, as you'll pay more and most require a special format only available on an IBM, on your TRS-80 you'll only come up with 20 meg anyway.

Now, how do you put it in? Very simple. To remove the old drive, you'll need to remove eleven screws, snip three wires and simply unplug the old drive.

If you don't care about the front panel WP switch and status light, you merely need to attach one of the snipped wires to five volts anywhere inside the case. If you want the lights and WP switch to work, three solder joints will be required.

This is a trivial installation for any hardware hacker, and certainly attainable by anybody that can wield a screwdriver, and can handle a soldering iron without burning themselves or what they point the iron at. If you don't feel up to it yourself, finding a local HH shouldn't be hard at all. Most computer clubs have one in residence, and with instructions from here they don't even have to be familiar with the TRS-80.

You will need the driver from Roy, as the Tandy driver only supports using 406 tracks, and most twenty meg drives have at least 612 tracks. Plus, the driver comes with a replacement for HARDCOPY/BAS that is much faster.

All your current programs should work unchanged.

Model I drive speed

Fm Michael Dauphin: Does anyone know how to detect the index hole in the floppy disk drive on a TRS-80 MOD I. I would like to write a program to test the speed of the drives. There used to be a program in one of the data areas, but it looks like they have been purged of most of the MOD I/III programs. If anyone has a public domain program that tests the rotational speed of the MOD I, that would be great.

Fm MISOSYS, Inc: Tim Mann wrote one in hybrid BASIC for a Model I. It was printed in the first issue of THE LDOS QUARTERLY (Vol I, Issue I, July 1, 1981). Send me a SASE and I will send a copy of the printed page.

FDC Alignment

Fm LDOS Support: To align the FDC board on a Model 3 or (some) Model 4 TRS-80 Computers:

(1) Determine what kind of FDC you have. If you have no separate FDC board, you have a 4P, or the gate array Model 4 desktop system. You are done, as the data separator and write precomp are all digital. No adjustment is possible. If it's broke, ya gotta fix it.

(2) Established that you do indeed have a separate FDC board, check to see if there are three screwdriver set potentiometers on the right had edge of the board. If not, you have the digital FDC board (rare in Model 3s unless you have the OEM Tandy 3 version, more common in Model 4s) and no adjustment is possible. See #1 above.

(3) Go for it. You will need the following equipment: (a) Digital Voltmeter; (b) Oscilloscope (at least 30 MHz will be barely adequate, > 50 MHz is better); (c) Frequency counter (optional); (d) assorted hand tools.

(4) Now we are ready to go to work. Disassemble the computer, and set the top with monitor aside to the left of the computer. Set it on its side so that it can still be connected and the computer can be turned on with the video operational. Remove the CPU shields. Remove all screws holding the CPU board. Remove the keyboard cable hold down screws from the right hand drive shield. Unplug the keyboard cable. That's it for disassembly.

(5) Now, place a piece of insulating material over the top of the upper disk drive. This is one of the few uses for a recent issue of 80micro. Carefully lift the CPU board to clear the bottom edge card connectors, then flip it up and around to rest on the above insulating material. Reconnect the keyboard cable, being sure to get pin one in the right place. You may have to pull out from underneath the drives to get enough slack.

(6) We are now set to actually run the machine with it disassembled. Boot something. When the drive has stopped, we can start adjusting. With the drive and FDC in their current "standby" state, we can make the first two necessary adjustments.

(7) Attach the DVM positive lead to test point twelve. If there are no staking pin test points installed on your board, use pin two of U14. Adjust pot R7 for a 1.40 reading on the DVM. Between 1.39 and 1.41 should be achievable by careful adjustment. Ground is available at test point four, or at many other spots on the board.

(8) Remove the DVM leads. Attach the frequency counter to test point thirteen. If no test point, use U14 pin seven, or U11 pin sixteen. If no frequency counter, use the scope carefully. Make sure that you pick exactly the same points of the wave form for measuring the period. Adjust R6 for a 2.0 MHz output (500 ns period).

(9) Remove the test leads. Attach the scope to test point eight. If no test point, use U13 pin one. Now comes the hard part. Position the drive to track thirty-nine, and issue continuous sector writes to a formatted scratch disk in the drive. Rotate R5 and watch what happens to the waveform. At one end of the rotation it will be very short, and gradually lengthen as you rotate R5. At some point, the waveform may seem to "collapse" back to a shorter length. If the point is not significantly past the 200 ns mark, like 250 ns or greater, replace the WD2143, U13. Otherwise, rotate back and leave the control set at 200 ns.

That's it! Detach the keyboard cable and reinstall the CPU board properly. You would normally proceed to a complete drive cleaning and alignment for each drive at this point.

DRAM refresh

Fm LDOS Support: No, don't get 4164 (REFRESH) chips, they aren't the correct type. What you need are 128 row in 2 microsecond chips, as opposed to 256 row in 4 microsecond chips. Motorola chips are an example of the correct variety; TI chips are an example of the incorrect variety. Jameco has the right kind for $7.95 per set of eight, or $14.95 if you want an old-style Model 4 PAL at the same time.

Fm Ralf Folkerts: On page 107 of TMQ I.iii is a question about refresh of dynamic RAMs. After looking into the Siemens 'Technisches Handbuch Bauelemente', reading a few articles

in magazines and talking with a few people I found a solution that seems to be good. First, the question at all is very good.

Everybody I talked to, knew that 4116 RAMs need 128 RFSH cycles while most 4164 and all 41256 need 256 RFSH cycles; when I asked 'why?' it got very quiet. Together with the manuals we got to the following solution (without warranty).

The RAMs consist of a condensator [(capacitor, C)] and a transistor. Because of leakages of the C, it has to be refreshed. The refresh should occur about each 2ms. The RAMs will normally store information longer, but the higher the temperature the higher the leakages will be. The 2ms is a 'worst case' value.

The RAMs are organized in a 128 by 128 matrix; therefore they need 7 address pins.

Only one row of the RAM has to be refreshed. All cells that are in that row will be refreshed then, too. To refresh the RAMs, the 7 bit column address has to be connected to the input pins of the RAM, then the RAS* signal must go active. Each row has to be refreshed all 2 ms.

The 'new' 64K and 256K chips are no longer organized in a 128 by 128 matrix. The 'normal' 64K chips are in 256 by 256 matrix. To refresh all rows there must be an 8-bit refresh address as 8 bits are necessary to address the column of the 256 by 256 matrix.

Some of the 64K chips are not organized in that 256 by 256 matrix but in a 128 by 512 matrix. Now there is a 7-bit row address but a 9-bit column address. One of the bits of the row address will be internally latched and used as a column address.

Update on 50 Hz

Fm Andrew Gransden: In answering a query from K W Arntsen in TMQ I.iii, you asked if anyone could answer the problem of keeping correct time with 50Hz. I developed a routine back in 1983 called TIME50/CMD, which was published in the 'LSI Journal' (Vol 2, #4, pg 222-26). This keeps my model III ticking to an accuracy of 3 seconds a day.

Basically the routine works as you describe by inserting itself in the interrupt chain and correcting the real-time clock by use of a couple of loop counters. The routine places itself in high memory and is SYSGENable in the normal way. The routine works on the Model III under both LDOS 5.1 and 5.3.

Fm Ralf Folkerts: re the 50 Hz question of K.W. Arntsen. In your answer you write that the interrupts in the III are based on the AC line frequency. That's not right. The first thing after R25 and the EMI filter is a bridge-rectifier - the whole power supply only gets DC 'power' ('geglattet' (filter ? (i don't know how to translate that)) by C4). The real time interrupts are caused by the VDRV clock processed through several gates.

When I got my III, I had the same problems - an inaccurate clock. The Model III TRSDOS 1.3I comes already configured for 50Hz clock - with a patch back to 60Hz. Therefore the clock ran fine until I started using LDOS. I came up with that problem by configuring the machine to 60Hz. I used that for almost one year now without a problem. That adjustment can be done by changing a few jumpers in the computer:

50 Hz	60 Hz
A-B	C-B
D-E	D-E
M-L	K-L
G-H	J-H
V-W	V-W
CC-BB	CC-BB

That works fine with 50Hz. It would be interesting to hear from other III users with 50 Hz.

[Also], In the LSI Journal 2.5, is a program that deals with the 50 Hz interrupts. Although I don't know why to use it (my 'jumpered' version is working fine) it may save you of writing a complete new version.

XLR8 and Single Density

Fm Gary Phillips To MISOSYS, Inc: While backing up the FixDisk and Utility disk just purchased from you, I discovered that my XLR8er enhanced 4P behaves badly in single density. It sounded familiar, so I checked, and sure enough you did mention a problem back in the Fall TMQ. Have you figured out what it is? Workaround (which you probably already know) is to SET180 the CPU for (M=3,I=4,R=10) ((which is S..L..O..W)) and then it can format/write/read single density. Is there a

timing loop or something in the LDOS FORMAT utility that goes too quick under XLR8er?

Fm MISOSYS, Inc: It's not the FORMAT utility, it's the FDC driver. Here's a little patch to allow DOS6 to work with SINGLE density media with an XLR8 board in and running at its fastest speed.

```
PATCH BOOT/SYS.LSIDOS (DOD,26=24:FOD,26=12)
```

The patch doubles the time delay after passing a command to the FDC. Works fine for me. Obviously, you will need to re-BOOT after applying the patch for it to take effect.

More on upgrading hard drives

Fm Mel Patrick: I had a look at the file HD1KGN.TXT and found it very interesting. I hadn't realized that the TRSHD6/DCT was actually a /CMD file until that time. Shows how one can assume things and be completely off base. However to add my own input (for your information and what its worth), I have a 5 meg RS HD (the big old black box original) and have been using it for the last two years. Recently the hard drive has been making more and more noise until I decided I would have a look at it. Seems the bearings in the 5 meg Tandon T602S have gone to that great bearing factory in the sky. I checked with Radio Shack about replacement of the bare 5 meg drive but that turned out to be more expensive than it was worth (and was expected from RS). My story does have a happy ending though. I have a friend who has an IBM with a 10 meg drive he has been using for about 6 months. He finally found that 10 meg on his IBM was pretty small and in his words "useless". So he bought a new 20 meg and I borrowed his old 10 meg to see if it was the same [as the 5meg]. The 10 meg is a CMI series and is identical in size and has the same connectors that the 5 meg did.

It looked like a case of straight swap so I did and was pretty surprised when I removed the 5 meg only to find RS has soldered 3 wires directly onto the 5 meg drive board. I removed the 3 wires which were soldered to U7 pin 5, U21 pin 8, and test pin 18 (at the back of the board). I installed the 10 meg drive but left off the three wires since I didn't know exactly where they would connect to the new board. I booted up LSDOS and used the usual SYSTEM (DRIVE=3,DISABLE,DRIVER="TRSHD6") and answered all the questions with exactly the same info as for the 5 meg but entered the number of cylinders as 306 (a 10 meg still has four heads, and usually the same step rate as the old 5 meg styles). It informed me that the drive appeared to be unformatted. I was getting pretty excited by that time so I tried formatting it. And it returned with the error, WRITE PROTECTED DRIVE. Humm....

After some probing, I find out that the wire RS had soldered to that test pin 18 at the back of the board was actually the write protect circuit. A logic 1 will enable writing, a logic 0 will disable (protect). I traced where the test pin 18 connected to. It goes to the J2 edge card connector foil pin 5. I located the same trace on the new board of the 10 meg drive and connected the same wire to it. I now had write protect solved.

I still haven't been able to figure out where to solder the remaining two wires for the active light.

From that point in time I formatted the drive and have been using it ever since with a full 10 meg.

I tried the idea of changing to the 1K sectors, but on 10 meg it only shows 1207K of storage out of 2448 available.

What I though you might be interested in was that if you want to double your storage, its not all that hard to install a 10 meg. I actually have another 10 meg here exactly the same as the first but for some reason if you connect it to the same controller board neither one will work. Which leads me to believe that RS has a controller board of some kind in their primary AND secondary drives. But with no schematics or info its like duck shooting in the dark. Thanks for the article though, it will provide a starting point when I have another look at the 1K sizes.

Fm Bob Haynes To Mel Patrick: The write protect harness plug on the 5 megger is wired as follows:

Pin 1	- White	- senses step status from drive (U7, pin 5)
Pin 2	- Yellow	- senses LED status from drive (U21, pin 8)
Pin 3	- Black	- "0" volts (ground)
Pin 4	- Purple	- Output to Active light ("0" volts when lit)
Pin 5	- Red	- +5 volts

The red lead provides a standing +5 volts to both the active and WP lamps. When the

"negative" side of the WP lamp is not tied to "0" volts via a pressed WP switch, WP is disabled by pulling TP18 high through the WP lamp filament. Not enough current is passed to light the bulb. Take note: if the bulb burns out, your drive is permanently write-protected! Strange!

The white and yellow leads go to a logic gate on the controller board which drives the active lamp (purple lead). Since you don't know where to pick up the CMI step signal, just tack the white lead to "0" volts.

On the 5 meg bubble, U21, pin 8 is the "drive LED" output. Of course, this is removed on Tandy bubbles, and the output instead tied to the controller and inverted (so the lamp stays on most of the time instead of off). You need to find the CMI LED output and connect the yellow lead there. The active lamp may behave normally, or may be "inverted" depending upon the output's polarity. I don't recommend connecting the active lamp directly; it is an incandescent bulb, not an LED, and the CMI's output chip may not handle the current well.

Whew, I'm rambling; does any of this make sense to you? Hope it helps! Oh, RS secondaries (at least 5/15 meggers) don't have controllers of any kind. What they do have is a little relay driven by, I think, +5 volts supplied by the primary. If the primary isn't connected, no 115 v goes to the power supply.

What you may have run into is an improperly "programmed" secondary; hard bubbles have drive select jumpers just like floppies. If your secondary was configured as "HD0", the controller would have become highly confused!

Fm Doug Mayfield: I have a question that could be a logical extension of the discussion going on the last few days about changing a 5 meg to a 10 meg hard drive. I have a 5 meg HD on my model 4 and am giving serious consideration to doing the upgrade to the 10 meg bubble (or even larger if anyone knows if that is possible, any ideas) that was previously discussed, however, does anyone know if it is possible to add another bubble to make a secondary (10 meg) onto the primary case. I know that there is more than enough room in the old RS case, and someone in an earlier letter said that which bubble is controlled by some sort of relay, but has anyone ever actually done this? While I don't really feel comfortable in doing this, I do have a couple friends that will do it for me if I can get the instructions. Also, a friend has a RS 15 meg HD on his model 4, would it be possible to do the same sort of thing in his HD case. Any ideas would be greatly appreciated.

Fm Bob Haynes To Doug Mayfield: That was my comment about the relay, Doug. The only reason for its existence is to prevent an RS 5/15M secondary unit (which consists of a bubble & p/s) from being powered up without being properly interfaced to a live controller. The cabling from the primary supplies +5 volts to operate the relay, which switches the 115v line. No primary? No +5v, no 115v power. All necessary because of the secondary's separate power cord.

Putting both bubbles in one cabinet eliminates the relay and makes things simpler, although all of Joe's previous comments certainly apply. I THINK with the right hardware and Roy's drivers you can get as much as 70-80 meg on-line.

More on 2-sided Floppies

Fm William Chao: I recently bought 2 TANDON TM 50-2 double-sided drives for my 4 and I would like to share my experience with others that may want to put in half-height DS drives into a 4. If your computer had the original drive tower from Tandy, you're in for a big surprise! The holes do not match exactly so that your new drives will either sit at the top or the bottom of the drive opening. I solved this by cutting out some cardboard and fill the open spaces. Not exactly a eye-pleasing sight but with Aluminum foils covering the cardboard, it does look a little better.

Second and MOST IMPORTANT!!!! The original Tandy screws for holding the drives in is TOO LONG! I didn't know that until I screw the screws all the way in and turned the power on. One drive just went up in a big cloud of smoke. What happened was the long screw hit the pc board and grounded-out the inductor that sits on the servo board. The dealer was nice enough to give me a new one even though I goofed! SOLUTION: Cut the screw to half of its original length.

Besides the new cables I had to make, putting in new drives was somewhat enjoyable and the Tandons are getting some heavy use besides my hard-drive.

The MAX-80 batteries - revisited

Fm Vincent Domeraski: Does anyone know whether the three internally installed batteries are available anywhere for the Max? If not, is there a reasonably simple alternative?

Jeff Brenton To Vincent Domeraski: The best I found at a "Mr. Nicad" booth is a 3-pack of 1/3AA cells. While it will NOT fit under the board like the original, it fits PERFECTLY 1/4" above the board! "Mr. Nicad" is:

E.H. Yost & Company
RR#1 Box #37
Sauk City, WI 53583
(608) 643-3194

I detailed the change procedure in a note that was published a couple of months ago in the MAXIMUL II newsletter, but it is fairly simple. The battery has 3 1/3AA cells taped and soldered together, with leads coming off for + and -. You unsolder the old battery from the bottom of the board, then turn it over. Observing polarity, you now solder the new battery in, spaced about 1/4" above the board to clear other components (the leads are about 3/4" long).

Having done that, you turn the Max on for at least 24 hours to charge the battery, and you are done. The pack costs $5 plus shipping and handling from Yost, or you can pick it up at a hamfest. If you are in the Midwest, look for a yellow "Mr. Nicad" sign at the larger 'fests.

Where to go - for standards

Fm Adam Rubin: The RS-232 "standard" is issued by the EIA. Apparently its full designation is ANSI/EIA-232-D-1986. (If anyone's looking for it, it's published by the Electronic Industries Association, Engineering Department, 2001 Eye Street, Washington, DC 20006, and goes for $20.)

I was just skimming through the library's copy, and didn't read everything thoroughly (about 50 pages), but I think they did specify which connector (male or female) goes on each piece of equipment.

XLR8 and Z80 equivalent speed

Fm Mel Patrick: Anyone have any idea what to set the XLR8er 64180 at to closely simulate a standard Z80. I definitely love the speed increase (I don't bother using it with the filters for max speed since I can live with 70% increase), but when writing applications it would be nice to return to more or less the same speed as everyone else's Z80.

Fm LDOS Support To Mel Patrick: Of the three, I/O waits isn't going to make much of a difference because you don't spend that much time doing I/O except when actually doing a sector read or write, and that is going to be most controlled by the interleave anyway.

Refresh is going to be a pretty constant background noise, and I wouldn't mess with that.

Lastly, memory waits is the biggie. Let's assume that a given instruction takes seven T-states to execute. I don't have my Z80 reference here, so I'm using hypothetical numbers. On the Model 4 with one M1 wait state this (data immediate, let's say) instruction would take eight T-states to execute. That's 250 ns times eight or 2,000 ns.

If your 64180 is clocked at just over 6 MHz, that makes a T-state about 165 ns. Those seven T-states now take about 1,155 ns, leaving about 845 ns to be eaten up in waits to match the old speed. That's five T-states that must be distributed over two byte fetches.

Since you can't put in part of a wait state, you could try adding three and see what happens on a CPU-intensive program (like BASIC). I'd guess at three rather than two because some of the 64180 instructions execute faster anyway. With a higher clock you'd have to add correspondingly more wait states. At 9 MHz you'd need at least five wait states.

XLR8 and Zeus

Fm Mel Patrick: Interesting things happen when you order a new piece of hardware. You sometimes find a program which will no longer work properly. In my case that was the XLR8er which uses the 64180 and has an additional 256K of RAM on board. I have only found one program which doesn't work on it. That being my favorite program, an editor assembler called Zeus. Granted there are many other editor assemblers with many other features but I like Zeus for

the speed of assembly. You know give something up to get something (some of my source files are in the 400K range). To continue Zeus just crashes when you have a XLR8er board installed. Seems the programmer who wrote Zeus used 1 undocumented op-code. In hex this is CB 31 and stands for SLL C which is exactly the same as SLA C, SET 0,C. I have located and removed these op codes since I suspect the 64180 only handles "documented" codes. If anyone wants the PATCH file, just leave me a note and I'll upload it.

On disk drive terminating resistors

Fm LDOS Support: A terminating resistor is installed on the floppy disk drive logic board, adjacent to where the 34 pin edge connector is connected [for an external disk drive]. There will be an empty socket that the terminator plugs into. I believe that on the TPI drives that the socket is a SIP (Single Inline Package) with one row of connectors instead of the normal DIP (dual inline package) with two rows of connectors like an IC.

A SIP terminator can be ordered from Radio Shack National Parts (817) 870-5662, probably as a part for a 26-1164 disk drive. You would have to specify a SIP terminator for a TPI drive, as that stock number was used both for the TPI (Texas Peripherals Inc) and Tandon-style drives that use the DIP terminator.

If you'd like to try to find one locally, count the number of pins needed to fit into the socket, and ask for a SIP resistor package, 150 ohms, with one less resistors than pins (i.e. a six pin package has five resistors in it), one common pin. If that is the right number of pins, that would be a:

Beckman	L06-1C151
CTS	770-61-R150
TRW	C06-1-151G
A-B	706A-151 (Allen-Bradley)
Bourns	4306R-101-151 or 4606X-101-151
Dale	CSC06A-101-151G

Any of the above would be fine for six pins, similar devices would be available with more pins and hence more resistors.

Configuring select on the drive with an all pins intact cable will require that you cut traces on the logic board.

The 64180 CPU

The HD64180/Z64180 CPU
by David Hall

The 64180 micro-processor manufactured by Hitachi (second sourced by Zilog) is a enhanced Z-80 compatible chip. There are several enhancements of both hardware and software, and some pitfalls to be aware of.

First I thought I'd discuss the enhancements (and pitfalls) that I have found in the hardware. The first thing that is noticed is the physical differences. The 64180 comes in three packages, a 64 pin DIP (what Hitachi calls a shrink dip), a 68 pin Flat Pack, and an 80 pin Chip Carrier; while the Z-80 is available in either a 40 pin DIP or a 44 pin Chip Carrier. My only experiences with either chip have been with the DIP packages. Right away, a problem arises for those of us who prototype by wire-wrap. The 64180's 64 pin package has 70 mil lead spacing, while normal wire-wrap boards and sockets are 100 mil spacing. At this time, there are no wire-wrap sockets for the 64180 that I have been able to find.

The 64180 also has several built-in peripherals. These include a crystal oscillator, an MMU, a 2 channel DMA controller, a 2 channel programmable timer, a 2 channel UART, programmable wait state and dynamic ram refresh generators, a clocked serial I/O channel, and an interrupt controller.

MMU

The internal Memory Management Unit allows the physical address space of the 64180 to exceed 64K. With the original 64180 (noted as R0 mask chips), the physical address space for the chip is 512K. With the newer chips (noted as R1 mask) the MMU has been expanded by one bit in the PC & FP packages, so that these chips have a physical address space of 1024K, while the DIP package chips are still limited to 512K. Since the 64180 has an expanded Z-80 instruction set, the logical address space of the 64180 (all types) is 64K. The tricky part (at least at first) is understanding the relationship between physical addresses, logical addresses, and how the MMU handles them through the three logical address areas.

The three logical address areas are called the common area 0, bank area, and common area 1. Common area 0 HAS to start at logical address X'0000' and physical address X'00000'. The bank area and common area 1 can both be set to begin on any 4K address boundary in the logical address space, with the requirement that the bank area be located at a lower logical address than common area 1 (if they both exist). The bank area and common area 1 are defined in two steps. First their location in the logical address space is specified. The starting logical address of the bank area defines the end of common area 0 in the logical address space. In a similar manner, the starting logical address for common area 1 also determines the ending logical address for the bank area. Secondly, the base address of the bank area and common area 1 may be freely set to begin on any 4K address boundary in the physical address space. My practice has been to define all three areas. I have used common area 0 to map any PROMs into the address space, common area 1 has been used to map RAM used for stack and common variable storage into the address space, and the bank area has been used to select any other area of memory that was needed. Typically once set up, only the bank area base address was changed to allow access to the full physical address space.

In a related topic, the 64180 has a 64K I/O address space. This is accomplished by presenting a full 16 bits on the address bus during any I/O instruction. If all I/O used by a system is addressed in the first page of the I/O map, then external I/O on the 64180 should be no different from I/O on the Z-80. When there are defined I/O ports > X'FF' though, I/O using immediate addressing will not work. This is because the 64180 creates the 16 bit address for an "IN (port),A" or "OUT A,(port)" instruction by placing the contents of A on address lines 8-15 while putting "port" on address lines 0-7. It is possible to load A with the proper value before doing an "IN (port),A" ... but not possible to do so for an output instruction in the general sense. When register indirect I/O is used, single byte I/O is simplified, but the auto-repeat I/O instructions then become useless. In the case where I/O is accomplished through the port addressed by the contents of register C ... such as OTIR; OUTI; INI; INIR; OUT (C),H or IN E,(C) ... the 16 bit I/O address is created by placing the contents of BC on the address bus. For the OUT (C),r and IN r,(C) instructions this provides a simple method of processing multi-byte I/O. However, when the auto repeating instructions are desired the situation becomes tricky once again. Remember that OTIR, OUTI, INIR, and INI all decrement register B, and these instructions also place BC on the address bus for port decoding!

DMA

The DMA controller has (so far) worked exactly as specified in the manual. Channel 0 can handle three types of DMA: Memory to Memory transfers, Memory to I/O (and I/O to Memory) transfers, and Memory to Memory mapped I/O (and Memory mapped I/O to Memory) transfers. Channel 1 handles only Memory to I/O (or I/O to Memory) transfers. Only Channel 0 is capable of using the on-board ASCI channels, but it is capable of using either channel as either the destination or source for DMA transfers, determined by how the destination (or source) address registers are programmed. Both channels have 20 bit memory address registers, allowing the DMA to directly control the full 1024K of address space (512K for the R0 packages); and 16 bit block count registers, allowing the DMA to directly transfer blocks up to 64K in length. In addition, the blocks can cross a 64K physical address boundary. An interrupt on DMA termination can be enabled when desired.

In the case of memory to memory transfers, the DMA allows both burst and cycle steal methods to accomplish the DMA transfer. In cycle steal transfers, the DMA controller and the CPU alternate control after each machine cycle (CPU) and DMA byte transfer (DMA). This alternate control of the data, address, and control busses will continue until the block has been transferred. In the case of burst transfers, once the DMA transfer has been initiated, the DMA controller will transfer the entire block without relinquishing control.

For memory to I/O transfers, the transfer rate is determined by the I/O device. /DREQ (which can be programmed for either level or edge sense) is used to signal the DMA controller when another transfer can take place. The transfer continues in a manner similar to the cycle steal method discussed above.

NOTE: In R0 chips, memory mapped I/O could only address memory in the bottom 64K of the address space. The R1 masked chips have been fixed to allow memory mapped I/O to any address in the physical address map.

Once a DMA transfer has been started, it can be disabled and enabled under control of the software (as long as a burst transfer was not used). Once the block has been transferred, further DMA is automatically disabled until once again started under program control.

My first direct experience with the DMA controller involved I/O to memory transfers using Channel 1 and an external receiver circuit. Using this approach, I was able to capture data at a rate of 187K bytes/sec (approximate) using a 64180 running at 3.072 MHz. Another application used DMA Channel 0 and an ASCI port to output data blocks at 19.2K bytes/sec while using an interrupt structure to support communications on three other 9600 bytes/sec serial channels.

ASCI

The two channel Asynchronous Serial Communications Interface provides two independent full duplex channels. Each channel can be programmed for 7 or 8 bit word length; 1 or 2 stop bits; odd, even, or no parity; data rate; and interrupt generation. Word length and parity are straight forward. Programming the data rate (using the internal baud rate generator) can be a little tricky and the rates available depend upon the clock frequency of the system. The technical manuals provide settings for frequencies of 6.144 MHz, 4.608 MHz, and 3.072 MHz. If you are using some other frequency, you may have to use an external clock source, and program the ASCI ports for external clocks, but this would then limit the use of DMA Channel 0.

The EIA control signals provided with the ASCI channels have not proved to be very useful in my opinion. Both channels provide for CTS input (although channel 1's CTS input is shared with another function for the CSI/O) which are unconditionally used to gate the transmitter ready function. I have found this to be quite limiting. Only Channel 0 has an RTS output (which I have used for a general purpose output), and a DCD input.

The ASCI interrupts do not operate quite as stated in the technical manuals. Specifically, it is stated that when receive interrupts are enabled, an interrupt will be generated when a rising edge of the DCD input is detected. It has been my experience that when receive interrupts are enabled, an interrupt is generated when the level of the DCD input is high (which would imply that the EIA signal was OFF). This interrupt request cannot be reset without disabling receive interrupts.

Finally, a word about 1/2 stop bits. On every UART, USART, SIO, or SCC I have ever programmed, when 2 stop bits are programmed, this has meant (and been documented) that the device will transmit characters with 2 bits times of mark at the end of each byte, but the device will still properly receive characters with only 1 stop bit. The ASCI ports are a little different! When 2 stop bits are programmed, not only will each data byte be sent with 2 stop bits, but each receive character must have 2 stop bits as well. Once I found this out (the hard way I might add), reading the manual "one more time" showed that the language in this area is not really clear, and could be taken either way.

PRT

The two channel Programmable Reload Timer provides two independent 16 bit down counters which are automatically reloaded when they overflow. Because the down counters can be directly read, it is possible to use them without enabling the interrupt capabilities. Their best use though would seem to be as software controlled interrupt driven interval timers. In addition, channel 1 has the ability to generate a software controlled wave form. It is possible to program TOUT to toggle when channel 1 is reloaded, which makes possible a square wave with software controllable frequency which does not require any interrupts, or other intervention once the PRT channel has been programmed. Since generating a waveform requires that A18/TOUT be used as TOUT, I have never used this feature of the 64180 chip. Each channel is capable of providing interval timing as fine as 20 cycles of the system clock.

NOTE: There was a bug in the original R0 chip which could allow spikes on the TOUT pin when address lines changed and TOUT was at a logic low level. R1 mask chips are reported to have corrected this problem.

Interrupt Control

The 64180 also has expanded interrupt capabilities, and an on chip interrupt priority encoder which operates in a manner similar to the Z-80 in interrupt mode two. The 64180 chip has four external interrupt inputs as well as 8 internal interrupt sources. The external interrupt inputs are NMI, INT0, INT1 and INT2. Operation of the NMI interrupt is

handled the same as on the Z-80; IEF1 is copied to IEF2, IEF1 is set to 0 (interrupts are disabled), the PC is pushed to the stack, and a jump to LOGICAL address X'0066' is performed. As with the Z-80, the final instruction of an NMI service routine should be an RETN to allow proper function of the interrupt enable flags. There is one additional action that an NMI causes in the 64180. DMA operation is suspended by clearing the DME bit in the DMA Control register (DCNTL). The NMI service routine can resume or abort an interrupted DMA operation, under software control.

INT0 functions in the same way as the INT input on a Z-80. The three interrupt modes which can be programmed for this input are the same as for the Z-80. The only change that I know of is that INT0 can be individually masked by a control bit in the INT/TRAP control register (ITC), as well as masked with IEF1 through the use of the DI instruction (along with the other maskable internal and external interrupts). INT0 has the highest priority of the external maskable interrupt inputs.

INT1 and INT2 operate in a manner that is similar to INT0 mode 2. The high order vector table address is supplied by the contents of the I register, while the low order vector table address is taken from the Interrupt Vector Low register (IL) which can be programmed to reside on any 32 byte boundary. Address bits 5, 6 and 7 are taken directly from the value stored in IL, while bits 0 - 4 are determined by which resource is requesting the interrupt. INT1 has a higher priority than INT2. Both INT1 and INT2 can be individually masked by control bits in the ITC, and are also controlled by the state of IEF1.

There are eight internal interrupt sources, seven of which are maskable. The TRAP interrupt is non-maskable, and is caused by in invalid op-code fetch. The TRAP interrupt response causes a jump to LOGICAL address X'0000'. The state of the TRAP bit in the ITC register can be used to distinguish between a RESET or RST 0 and a TRAP response.

The seven maskable interrupt sources are all individually maskable (by control bits in their separate control registers) as well as controlled by the EI, DI instruction pair. All seven interrupts respond with a vectored response mode which uses the IL register to provide the low address of the vector table. The I register provides the high order address, just as it does for INT0 mode 2 interrupts, and for the INT1 and INT2 interrupt responses. The internal interrupts are (in decreasing order of priority): PRT channel 0, PRT channel 1, DMA channel 0, DMA channel 1, CSI/0, ASCI channel 0, and ASCI channel 1.

NOTE: If an NMI request occurs while the CPU is fetching an undefined op-code, there is the possibility that the 64180's response will be a jump to logical address X'0066' rather than X'0000' (as the response to the TRAP interrupt request). When this happens, the TRAP service routine will not be executed, and the NMI service routine will be executed twice (the NMI request input will be sampled just after the jump to X'0066' and will cause a second jump to logical address X'0066'. It has been suggested that if the op-code as logical address X'0066' is changed to a NOP, then this situation can be detected and corrected though proper implementation in the NMI service routine. Specifically, at the start of the NMI service routine, the TRAP bit in the ITC register should be checked. If it is not set, then normal operation of the NMI routine can continue. If the TRAP bit is set, then the current top of stack needs to be checked for a value of X'0067'. If the top of stack is other than X'0067', normal NMI handling can continue, but if the top of stack is found to be X'0067', it should be changed to X'0000', so that once the NMI service routine has finished, the TRAP routine can properly handle the undefined op-code fetch.

I have not mentioned the Clocked Serial I/O port (CSI/0) because I have had no occasion to use this resource of the 64180 chip. Therefore I have nothing to offer, other than to try and paraphrase the technical manuals. Since I do not know that the CSI/0 works as documented, I have chosen to leave this area of the 64180 for others to discuss.

XLR8 Your 4P/III

Model 3 Mode, the 4P, and XLR8er

by Gary Lee Phillips

Some folks, myself included, consider Tandy's Model 4P microcomputer to be the most flexible and powerful Z80-based system ever sold to mere mortals. The addition of an XLR8er expansion board from H.I.Tech Inc. can really turn the 4P into a "mean machine" capable of outshooting most MS-DOS systems currently in use.

However, if you use the model 3 mode of your 4P very much, there is a small nuisance about installing XLR8er. The XLR8er replaces your Z80 CPU with the more advanced HD64180. The HD64180 can address more memory and run at higher speeds under software control (it does not require changing the actual clock rate, unlike the Z80 design of the models 4 and 4P). When you reset or first apply power to the XLR8er system, it defaults to its slowest possible speed, which is somewhere near the speed of the old model 1 TRS-80. The boot ROM in a model 4P makes certain assumptions, including a CPU clock rate of 2.0 MHz or thereabouts, which results in a serious problem: if you try to cold boot the XLR8er into model 3 mode, it will usually fail while trying to read the ROM image file from disk. The typical messages will be "Lost Data Error" or "ROM Image not found on Drive 0."

There are various ways to get around this. The one recommended to me by H.I.Tech was to load the ROM image after booting in model 4 mode. You can use the SET180 command to speed the machine up to normal or faster speed, then use the BOOT command of LS-DOS 6.x. Hold down the "P" and "F3" keys, and the machine will load MODELA/III from your LS-DOS 6.x system disk, and then prompt you for your LDOS or TRSDOS 1.3 master disk followed by enter. This usually works, though sometimes the machine comes up with the "Cass?" prompt instead of reading the model 3 master diskette. (I recover from this situation by pressing enter twice to get to ROM Basic, then using a SYSTEM command, followed by /0 and enter. LDOS will then initialize.) Hardly a friendly solution, right? But it is tolerable if you don't use model 3 mode very often. Unfortunately, I support a bulletin board system that runs in model 3 mode because it was originally set up on a real model 3. While developing and testing code for this system, I frequently have to reboot after crashing the DOS. Swapping disks and booting LS-DOS, then rebooting and reloading the ROM image each time gets downright irritating.

With a little encouragement from Roy and others on Compuserve, I set out to find the reason for the XLR8er's inability to read MODELA/III cold. The problem turns out to be merely one of speed: the code in the 4P boot ROM is not fast enough to keep up with a double-density diskette when the CPU is running in its slowest mode. Obviously the answer is to get the HD64180 kicked up to a higher speed before trying to load MODELA/III. This could be done by modifying the boot ROM (i.e., patch the code and burn it onto a new EPROM, open the 4P and swap chips...not a simple job), or by patching the boot sector of LDOS 5.x to include an initialization routine. The latter seemed reasonable as an approach until I realized that MODELA/III has to be loaded before the boot sector gets control. It would be necessary to make the boot sector look like a model 4 boot sector, initialize the CPU, and then boot again, this time fooling the ROM into thinking it needed MODELA/III, even though the boot sector looked like model 4 mode.

Finally it occurred to me that the 4P has the ability to load "customized" ROM images. I haven't seen much written on the subject, but it is documented in the 4P Service Manual from Radio Shack. By default, the system will load the first file it finds in the directory with a name of MODELx/III where the "x" can be any valid character. If you hold down any key from "A" through "G" during the reset, the "x" will be replaced by the specified key, and that specific file will be loaded. Why not write a very short initialization program, and put it on the disk with a name like MODELB/III? Hold down the "B" key when you reset the computer, and let the MODELB/III file load and execute. Since it is very short (only one sector) the odds are much better that it will load without error even with the slowed-down CPU. This program can then speed up the XLR8er and reboot, letting the default MODELA/III file be loaded instead. Thus, XBOOT was born.

This short program can be assembled by any model 4 or model 3 Z80 assembler. Be sure to generate a /CMD file rather than a /REL file if your assembler has relocating capability.

Copy the output file to your LDOS master diskette (not the original, of course, but a working backup) giving it the name MODELB/III. Make sure MODELA/III is also present on the disk. Now put the INSXBOOT/JCL file on the same LDOS diskette. Get booted up in LDOS, using whatever method you have to in order to get the ROM image loaded this one time. Using the disk with MODELB/III and INSXBOOT/JCL as your system diskette, issue the command: DO INSXBOOT (D=1). When you are prompted, insert a scratch diskette into drive 1, and INSXBOOT will build a new master copy of LDOS, with the MODELA/III and MODELB/III files in the best possible directory slots for successful loading. Since the boot ROM reads the directory of drive 0 sequentially, the farther the MODELx/III file is down the directory list, the better the chances are of getting a lost data error before it can be found or read. INSXBOOT puts a temporary patch on SYS2/SYS to force these files into the top of the directory, rather than letting them fall into random positions. Once you have built a master diskette using INSXBOOT, you can use QFB to copy it as often as you wish and these crucial files will remain in the proper positions. The LDOS or LS-DOS BACKUP utility will also keep the files in place, but only if you are doing mirror image backups. Selective backup or reconstructs will redistribute the files randomly in the directory, so beware.

To boot using the new master diskette, put it into drive 0. Hold down the "B" key on the keyboard, and press the reset button. If you were already running in model 3 mode, you will also need to hold down the "L" key this time to force the ROM image to be reloaded. If you get a "Lost Data Error" message, just try again. It usually works on the first or second try. Once MODELB/III is loaded, it will prompt you to press enter again. At this point, you can use any added key presses desired (see your model 4P manual for details) after pressing enter, and MODELA/III should load successfully and pass control to LDOS.

If instead, you find MODELB/III has loaded a second time, the files are in the directory in the wrong order. MODELA/III must appear first on the directory (you can check this by using DIR :d (O=N) to get the unsorted directory listing). If you ran INSXBOOT, it will have taken care of this for you.

Now what, you may ask, about TRSDOS 1.3? I hope none of our readers are still dependent on this very weak DOS for their model 3 mode operations. However, TRSDOS 1.3 does have one or two useful features. In particular, I have used it when I had to get files from a model 1 diskette without destroying that diskette (remember, if you use the REPAIR command of LDOS or LS-DOS, the model 1 can't read the diskette any more). You can do a two-step conversion, moving the files first to TRSDOS 1.3 and then to LDOS or LS-DOS, and the original diskette will remain unchanged.

However, TRSDOS 1.3 has some definite problems with the XLR8er. Besides the same inability to cold boot that we just solved for LDOS, TRSDOS 1.3 contains instructions that will not execute correctly on the HD64180 CPU, causing the system to reboot when you try to execute a program, or use the BACKUP command. These instructions are of the type that "split" the index register IX or IY, loading the two halves separately. They are not standard Z80 instructions, but they do work on every Z80 chip I've ever tried them on. Tandy apparently used them in TRSDOS 1.3 as a "security" measure to make it harder to disassemble. I am aware of two places in TRSDOS 1.3 that contain these instructions. The following patch commands will eliminate them (note that this is a TRSDOS 1.3 patch, and NOT an LDOS patch, so please execute it under model 3 TRSDOS!)

```
PATCH *2 (ADD=4EBF,FIND=FD2EFFE1,CHG=FD21FF42)
PATCH *2 (ADD=4EC3,FIND=F1F1FD2642,
          CHG=E1F1F10000)
PATCH *7 (ADD=5284,FIND=DD2662DD2E24,
          CHG=000000000000)
PATCH *7 (ADD=528A,FIND=DD7E00,CHG=003EFF)
```

The patch to SYS7, by the way, also eliminates the limited backup feature of TRSDOS 1.3. Once these patches have been applied, TRSDOS 1.3 appears to execute normally on a 4P equipped with XLR8er. If you find other places where a patch is required, please let us all know!

If you want to boot a TRSDOS 1.3 diskette cold, requiring a load of the ROM image file, you will need to get both MODELA/III and MODELB/III onto the diskette so that they appear at or near the top of the directory, with MODELA before MODELB. Since TRSDOS 1.3 does not allocate directory entries randomly, this is easy to achieve by deleting the top files and then using CONVERT to move the MODELx/III files onto the disk from a 35-track SSSD LDOS diskette. Then copy the other files you deleted back onto your diskette from a backup copy. The TRSDOS 1.3 BACKUP command always produces a mirror image, so once you

get the files into the right position, BACKUP will keep them there.

Notes on the XBOOT program code:

The HD64180 has additional instructions that can be used to address its internal control registers. However, these instructions do not have to be used. The ordinary Z80 instruction, OUT (C),A or IN A,(C) will address the internal registers IF the value in the B register is zero. I have taken advantage of this fact in order to write an initialization routine that can execute harmlessly on a regular Z80, so if you happen to boot your special XLR8er system diskette on a 4P that does not have the HD64180 installed, it will still work. This diskette will also boot normally on a regular model 3 or a model 4, which would not have been the case if I had chosen to patch BOOT/SYS instead of building the special MODELB/III file.

When XBOOT (MODELB/III) gets control, it sets the dynamic memory refresh rate to every 40 t-states rather than the default value of 10. It also sets two wait states on every memory access, and two wait state on every I/O port access. This produces an effective CPU speed slightly faster than the standard model 3. If you wish to experiment with other speeds, you can change the values that are loaded to the RCR and DCNTL ports appropriately. You will need Hitachi publication #U77, HD64180 8-Bit High Integration CMOS Microprocessor User's Manual, in order to select the appropriate values. You can obtain this useful book from H.I.Tech or from Hitachi. H.I.Tech also has a version of SET180 that executes under LDOS. (I wrote my own instead, using the disassembly of the TRSDOS/LS-DOS version as a model.) XBOOT also removes the write protection from the ROM image area of RAM, and pokes a zero value into 3000H. This is one of the addresses checked by the boot ROM to make sure that the ROM image is not already present, and since it is not a C3H as expected, a full reload of MODELA/III is forced. XBOOT then pages in the boot ROM by loading a 01H value to port 9CH. Since the boot ROM was already executing when XBOOT received control, the necessary RAM areas are still properly initialized, and XBOOT can make use of boot ROM functions, such as the message display function accessed via RST 18H. These functions are documented in the model 4P service manual. After the CPU is initialized and the user gives the go ahead by pressing enter, control is passed to the boot ROM, which then will attempt a load of the "real" ROM image, MODELA/III.

Notes on INSXBOOT/JCL

Note that INSXBOOT performs a temporary patch to SYS2/SYS. This patch works on LDOS 5.3 only. If you are installing on an earlier version of LDOS, you will have to make sure you have a correct modification of the patch. Likewise, a similar patch could be made to LS-DOS/TRSDOS if desired, but the patch would not be identical. LDOS 5.3 normally selects a starting place in the directory for creating a new file by looking at the clock heartbeat. The result is random distribution of files across the available directory entries, which is generally desirable. However, in our case we want to get MODELA/III and MODELB/III into the very first available directory record. INSXBOOT insures this by forcing the selection to a hard-coded offset for each of these two files, and then restores the original algorithm to SYS2. Please don't run INSXBOOT on your master LDOS diskette! Use a backup copy, just in case the patch causes problems.

Thanks to Roy Soltoff, Joe Kyle Di Pietropaolo, and Adam Rubin for valuable suggestions and bits of information that went into this project. I'd be happy to hear about any improvements, or to try to assist you with any problems you experience with the code. I can be reached on Compuserve's LDOS forum, and my user identification is 72425,354.

```
. INSXBOOT -- this JCL will build an LDOS 5.3 system disk
. in the target drive (default is drive :1) such that
. the MODELA/III and MODELB/III files are placed in the
. first sector of the directory. To execute this procedure
. you need a backup copy of your master LDOS 5.3 disk in
. drive 0, with the two ROM image files copied onto it.
. Invoke with DO INSXBOOT (D=drive) where 'drive' is
. the number of the target drive. The optional parameter
. (DS) can be used if the target drive is double-sided.
.    -Gary Phillips, July, 1987.
.
//IF DS
//ASSIGN S=2
//ELSE
//ASSIGN S=1
//END
//IF D
. Target drive is :#D#.
//ELSE
//ASSIGN D=1
. Default drive :1 will be used.
//END
//PAUSE Insert blank disk into drive :#D#, <Enter> when ready.
format :#D# (sides=#S#,name="LDOS-530",mpw="PASSWORD",dden,abs)
backup sys0/sys:0 :#D# (s)
backup sys$/sys:0 :#D# (s,new)
. Now installing temporary patch to SYS2/SYS
patch sys2/sys.system:0 (d03,d4=3e 40 00:f03,d4=3a 88 42)
backup modela/iii:0 :#D# (i)
patch sys2/sys.system:0 (d03,d4=3e 60 00:f03,d4=3e 40 00)
backup modelb/iii:0 :#D#
. Removing temporary patch to SYS2/SYS
patch sys2/sys.system:0 (d03,d4=3a 88 42:f03,d4=3e 60 00)
. Copying remaining files to target disk
backup $:0 :#D# (s,i,new)
. Installation completed. To boot LDOS 5.3 with XLR8er 4P,
. use new system disk in drive :0, and hold down "B" while
. pressing the RESET button. If you get a "Lost Data Error"
. just try again. It usually works within two tries.
. To create additional copies of the new disk, you can use
. QFB to avoid running this process again.
//EXIT
```

```
MISOSYS MRAS-1.0 07/12/87 13:35:24 XLR8ER LDOS BOOT            Page 00001

00002 ;-----------------------------------------------
00003 ; XBOOT -- assist program to initialize the
00004 ;       HD64180 CPU in an XLR8er board before
00005 ;       attempting to load the MODELA/III file
00006 ;       on a 4P.  Assemble this code, and rename
00007 ;       it to MODELB/III.  Copy it to your boot
00008 ;       diskette for LDOS or TRSDOS 1.3.  When
00009 ;       booting "cold" into model 3 mode, hold
00010 ;       down the B key when you press reset. Then
00011 ;       when you are prompted to press <Enter>,
00012 ;       do so and proceed with normal booting
00013 ;       procedure, including any special keypress
00014 ;       you may require.
00015 ;
```

```
                  00016 ;       For best results, the directory entry for
                  00017 ;       this file should be in the first directory
                  00018 ;       sector, immediately after that for MODELA/III.
                  00019 ;       This can be achieved by zapping DIR/SYS or
                  00020 ;       by using the INSXBOOT/JCL file.
                  00021 ;
                  00022 ;       Gary Lee Phillips, CIS ID [72425,354]
                  00023 ;       Midsummer, 1987
                  00024 ;-----------------------------------------------
                  00025 ; Define the ports we need to use:
                  00026 ;-----------------------------------------------
0032              00027 DCNTL   EQU     32H             ;dma/wait cntl port
0036              00028 RCR     EQU     36H             ;refresh cntl port
0084              00029 OPREG   EQU     84H             ;memory mapping cntl
009C              00030 BOOT    EQU     9CH             ;boot rom cntl port
                  00031 ;-----------------------------------------------

8000              00033         ORG     8000H           ;above everything
8000              00034 XBOOT   EQU     $               ;else that we use
8000  F3          00035         DI                      ;just to be safe

                  00037 ; Set up the internal 64180 port values:

8001  0600        00039         LD      B,00H           ;simulate in0/out0
8003  0E36        00040         LD      C,RCR           ;but z80 compatible
8005  ED78        00041         IN      A,(C)           ;get refresh cntl
8007  F602        00042         OR      02H             ;CYC1 = 1
8009  E6FE        00043         AND     0FEH            ;CYC0 = 0
800B  ED79        00044         OUT     (C),A           ;set 40 state refresh
800D  0E32        00045         LD      C,DCNTL         ;now get wait state
800F  ED78        00046         IN      A,(C)           ;cntl bits and set them
8011  F690        00047         OR      90H             ;MWI1 = 1, IWI0 = 1
8013  E69F        00048         AND     9FH             ;MWI0 = 0, IWI1 = 0
8015  ED79        00049         OUT     (C),A           ;2 mem waits, 2 i/o

                  00051 ; Set up the 4P memory control ports:

8017  3E01        00053         LD      A,01H           ;remove memory write
8019  D384        00054         OUT     (OPREG),A       ;protection
801B  AF          00055         XOR     A               ;make sure ROM image
801C  320030      00056         LD      (3000H),A       ;gets (re)loaded
801F  3E01        00057         LD      A,01H           ;get boot ROM resident
8021  D39C        00058         OUT     (BOOT),A        ;in low memory

                  00060 ; Use boot ROM routine to display banner message:

8023  B7          00062         OR      A               ;reset Z flag bit
8024  219E80      00063         LD      HL,MVECT        ;point to msg block
8027  DF          00064         RST     18H             ;display prompt

                  00066 ; Now wait for user to press <Enter>:
                  00067 ; (If you prefer not to have to press a key to continue
                  00068 ; the bootstrap operation, omit the next 3 instructions.

8028              00070 KSCAN   EQU     $
8028  3A4038      00071         LD      A,(3840H)       ;read kbd matrix
802B  CB47        00072         BIT     0,A             ;enter key?
802D  28F9        00073         JR      Z,KSCAN         ;no, so wait

802F  C30500      00075         JP      0005H           ;do normal boot
```

```
8032  58      00077 MSG1    DB      'XLR8er Boot Assist '
8045  76      00078         DB      'version 1.0',00H
8051  20      00079 MSG2    DB      '  by Gary L. Phillips,'
8067  20      00080         DB      ' June, 1987.',00H
8074  50      00081 MSG3    DB      'Press <ENTER> to '
8085  63      00082         DB      'continue with bootstrap.',00H

809E  4000    00084 MVECT   DW      1*64,MSG1,-2
80A4  8000    00085         DW      2*64,MSG2,-2
80AA  0001    00086         DW      4*64,MSG3,-1

8000          00088         END     XBOOT


@@1           0000  @@2           0000  @@3           0000
@@4           0000  BOOT          009C  DCNTL         0032
KSCAN         8028  MSG1          8032  MSG2          8051
MSG3          8074  MVECT         809E  OPREG         0084
RCR           0036  XBOOT         8000
00000 Total errors
39884 Free space
```

The Patch Corner

General Information

The following information should be read before you type into a file, any of the patches noted in THE MISOSYS QUARTERLY.

It is unfortunate that our printer prints the letter "O" and the number "0" almost identically. Unless we utilize a filter to "slash" the number zero, the two are difficult to distinguish. However, when it comes to patches, all is not lost. In an LDOS 5 or TRSDOS 6 direct patch, the letter "oh" is not used in the patch code (it may appear in comments which are lines beginning with a dot). The direct patch format of TRSDOS 6 which we use in our patches is:

```
Drr,bb=xx xx xx xx xx xx ...
Frr,bb=xx xx xx xx xx xx ...
```

The patch is usually a pair of lines. The first line begins with the capital letter, "dee". This is immediately followed by the "rr" field (which stands for record). The "rr" field is always two hexadecimal digits. Actually, it can be a 4-hexadecimal digit number if the file to be patched has more than 256 sectors. Hex digits use nothing but the numbers zero through nine and the first six letters of the alphabet: A,B,C,D,E,F, or a,b,c,d,e,f. The record number is immediately followed by the "bb" field (which stands for byte). The byte field is also two hexadecimal digits - just like the record field. This is immediately followed by an equal sign, "=". The equal sign is immediately followed by the first patch byte (the "xx" shown above). The patch byte is again two hexadecimal digits. Where more than one patch byte is included on a line, it is separated from its predecessor by a single SPACE. The line is terminated with an ENTER.

TRSDOS 6 and LDOS 5.3 patch formats use a "find" line record. This is used to verify that the file being patched is actually the file you want patched. All of the bytes noted in the "F" line or lines must be matched in the file before any of the "D" patches will be utilized. The second line of the pair begins with the letter "F" which stands for FIND. The next six positions are identical to the preceding "D" line. Following the equal sign on the FIND line are pairs of hexadecimal digits which should align themselves with the preceding line.

So far, the letter "oh" is not used. The only place outside of a comment line where you could find the letter "oh" used is if instead of showing the patch bytes as a series of hexadecimal pairs, it was depicted as a string. A string could be used if one was patching a string of displayable ASCII characters. For instance, the patch:

```
D03,14="This is a new string"
F03,14="extra space for what"
```

would replace the string, "extra space for what", with the string, "This is a new string". Strings are shown within double quotes. That's the only place where a letter "G" through "Z" could be used.

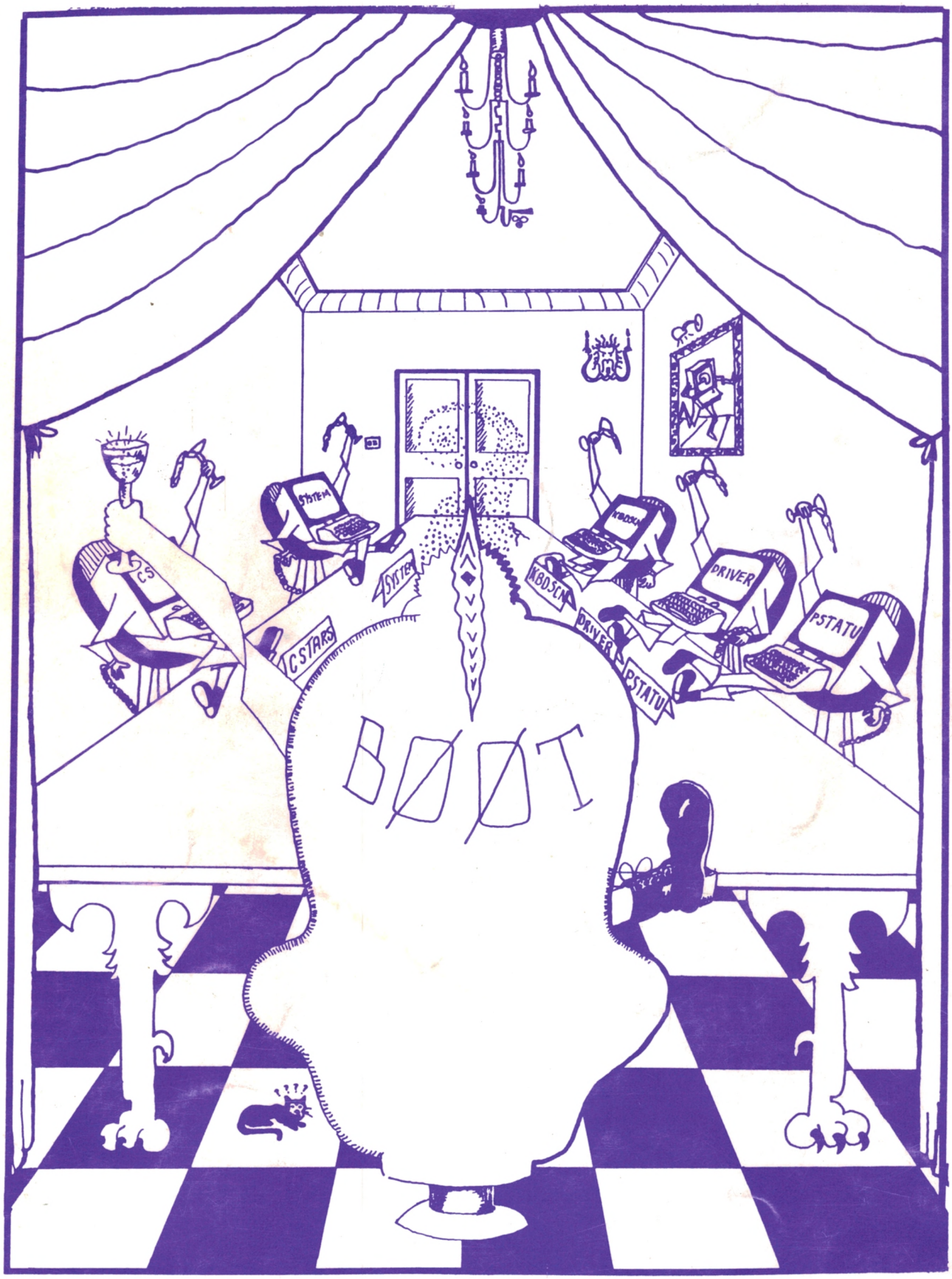
Also, even though TRSDOS supports the colon notation to put more than one patch line on the command line (e.g. "PATCH TEST (D01,27=56:F01,27=65)"), it does not support the colon separator when used in a FIX file (it does support a semicolon which is used under LDOS to signify a trailing comment); LDOS 5.3 supports a colon separator both in a command line patch and a fix-file patch. In order to conserve space in THE MISOSYS QUARTERLY, we may logically print more than one FIX line on a printed line; HOWEVER, ALWAYS USE A HARD <ENTER> FOR THE COLON WHEN TYPING IN A FIX FILE for TRSDOS 6.

If you use the FIXES?/TXT file from the DISK NOTES corresponding to this issue, please separate out the individual fixes which you need by use of any text editor you find convenient to use.

```
. BADBL51/FIX - fix error reported by Shane Dawalt - MC51 - 08/01/87 - RND
D2F,0A=CD 9F AF:F2F,0A=CD 06 AC:D5F,0A="fp operation":F5F,0A="floating poi"
D5F,16=00 21 02 00 39 E5 CD A9 92 F1 C3 06 AC:F5F,16=6E 74 20 6F 70 65 72 61 74 69 6F 6E 00
. Eop
=====
. BADBL60/FIX - fix error reported by Shane Dawalt - MC60 - 08/01/87 - RND
D39,01=CD 98 9B;F39,01=CD 7E 90;D77,63="fp operation";F77,63="floating poi"
D77,6F=00 21 06 00 39 E5 CD 93 74 F1 C3 7E 90;F77,6F=6E 74 20 6F 70 65 72 61 74 69 6F 6E 00
. Eop
=====
. BADBL61/FIX - fix error reported by Shane Dawalt - MC61 - 08/01/87 - RND
D2E,F9=CD 8E 83;F2E,F9=CD F5 7F;D5E,F9="fp operation";F5E,F9="floating poi"
D5F,05=00 21 02 00 39 E5 CD 98 66 F1 C3 F5 7F;F5F,05=6E 74 20 6F 70 65 72 61 74 69 6F 6E 00
. Eop
=====
. BRINGUP1/FIX - patch to PRO-WAM's 2.x BRINGUP/APP
. Corrects <BREAK> during text ADD
. Apply via, PATCH WAM1/APL BRINGUP1 & PATCH WAM0/APL BRINGUP1
D29,59=68 2F;F29,59=B4 2C
D2D,68=CD B4 2C D0 36 00 C9;F2D,68=00 00 00 00 00 00 00
. Eop
=====
. CAL1/FIX - Patch to PRO-WAM's 2.x CAL/APP
. Corrects display of "today's date" when switching years
. Apply via, PATCH WAM1/APL CAL1 & PATCH WAM0/APL CAL
D02,C5=4D;F02,C5=71;D06,E8=70 2E;F06,E8=8F 2C
D08,35=47 6F 74 6F 20 20 20;F08,35=42 72 69 6E 67 75 70
D08,3D=4E 65 78 74 20 20 20;F08,3D=20 20 20 47 6F 74 6F
D08,45=50 72 65 76 20 20 20;F08,45=20 20 20 4E 65 78 74
D08,4D=42 72 69 6E 67 75 70;F08,4D=20 20 20 50 72 65 76
D08,6D=3F 20 03 CD 8F 2C 3E C9 32 DE 2C C9;F08,6D=20 3F 20 03 62 72 69 6E 67 75 70 03
. Eop
=====
. CAL2/FIX - Patch to CAL/APP of PRO-WAM 2.x
. Fixes display of stars in leap year and on the 31st of a month
. Apply via, PATCH WAM0/APL CAL2 & PATCH WAM1/APL CAL2
D04,39=5E 23 56 21 8D 2E 19 FE 03;F04,39=FE 03 5E 23 56 21 8D 2E 19
D04,45=63;F04,45=5F;D07,18=63;F07,18=5F;D07,39=63;F07,39=5F
D07,5A=7E E6 1F 47 23 7E E6 F0 FE FF 20 22 7E E6 0F 87 C6 0E 4F FE 14 38 08 3A
F07,5A=23 7E E6 F0 FE FF 20 26 7E 2B E6 0F 87 4F 06 00 FE 06 38 08 3A 5F 2D E6
D07,72=63 2D E6 30 20 01 04 26 24 69 78 86 23 4F 8E 91 47 21
F07,72=30 20 01 34 7E E6 1F 21 0E 24 09 4F 7E 23 66 6F 09 01
. Eop
=====
. CED52/FIX - Patch to EnhComp's CED/CMD
. Corrects "X" command when the BASIC line contains a "/" character
. Apply via, PATCH CED CED52
D05,A6=C5 6A:F05,A6=C4 5B
D14,2A=7E 12 B7 C8 23 13 C3 C5 6A:F14,2A=00 00 00 00 00 00 00 00 00
. Eop
=====
. CED63/FIX - Patch to PRO-EnhComp's CED/CMD
. Corrects "X" command when the BASIC line contains a "/" character
. Apply via, PATCH CED CED63
D05,B2=DA 3E;F05,B2=D0 2F
D14,3F=7E 12 B7 C8 23 13 C3 DA 3E;F14,3F=00 00 00 00 00 00 00 00 00
. Eop
=====
```

```
. DDFORM53/FIX - Patches to diskDISK DDFORM/CMD - 06/30/87
. These patches change DDFORM to format using LDOS 5.3
. dating conventions. Based on a fix supplied by M. Pollard
. Apply using "PATCH DDFORM DDFORM53"
D02,A7=53:F02,A7=51:D03,52=00 00:F03,52=F5 9C:D03,72=00 00:F03,72=96 42
D03,D4=CD 15 5D:F03,D4=32 9D 54:X'5D15'=F6 88 32 9D 54 C9
. Eop
=====
. DDFORM63/FIX - Patches to LS-diskDISK DDFORM/CMD - 06/30/87
. These patches change DDFORM to format using LS-DOS 6.3
. dating conventions. Based on a fix supplied by M. Pollard
. Apply using "PATCH DDFORM DDFORM63"
D02,A5=63;F02,A5=52;D02,B8=4F;F02,B8=47;D03,7D=00 00;F03,7D=F5 9C
D03,9D=00 00;F03,9D=96 42;D03,FF=F6 88;F03,FF=CB FF
. Eop
=====
. DDFORM6A/FIX - Patch to LS-diskDISK DDFORM/CMD - 06/24/87
. This patch corrects the problem of DDFORM placing the Media Data Block
. in the GAT sector at GAT+X'F4', instead of GAT+X'F5' where it belongs.
. Apply using "PATCH DDFORM DDFORM6A
D02,DF=F5;F02,DF=F4
. Eop
=====
. MFX51/FIX - Patch to MRAS FIXUP - 08/03/87
. Corrects fixup of files which have NULL lines
. Apply via, PATCH FIXUP (D01,15=00 00:F01,15=2B 2B)
=====
. MFX61/FIX - Patch to PRO-MRAS FIXUP - 08/03/87
. Corrects fixup of files which have NULL lines
. Apply via, PATCH FIXUP (D01,06=00 00:F01,06=2B 2B)
=====
. PSORT22A/FIX - Patch to PSORT 2.2a - 08/24/87
. Corrects minor errors
. change version to 2.2b
D02,60=62;F02,60=61
. allow sort of full record
D07,36=00;F07,36=23;D07,51=BD;F07,51=B7
. add default extension to prompted input
D04,40=ED 36;F04,40=7B 57
D0F,D2=E5 21 1F 26 7E 23 B7 28 06 D6 0D 20 F7 2B 77 E1 C3 7B 57
F0F,D2=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.EOP
=====
. SYS5A/FIX - Patch to LDOS 5.3 SYS5/SYS - 06/29/87
. Adds length record for SYSTEM (SYSRES=5)
. Apply via, PATCH SYS5/SYS.SYSTEM SYS5A (O=N)
D04,42=01 04 C8 4B FF 03 02 02 00 4E
. Eop
=====
. SYS6F/FIX - 07/22/87 - Patch to LDOS 5.3 DIR command
. re: bug in abort exit from JCL referred by J. Hawes
. Apply via, PATCH SYS6/SYS.SYSTEM (D07,0D=30 40:F07,0D=28 5A)
. Eop
=====
. SYS6G/FIX - Patch to LDOS 5.3 DEVICE command
. Corrects detection of hardware WP for status display
. Apply via, PATCH SYS6/SYS.SYSTEM SYS6G
D2C,C1=7F:F2C,C1=77
. Eop
=====
```

```
. SYS7F/FIX - Patch to LDOS 5.3 SYS7/SYS - 06/29/87
. Corrects fixup on "SYSTEM (SYSRES=n" with bad "n"
. Apply via, PATCH SYS7/SYS.SYSTEM SYS7F
D11,91=2F:F11,91=2E:D11,99=27:F11,99=26:D11,9D=23:F11,9D=22
D11,BC=F1 21 4F 5A DD 21 31 5A CD 8A 42:F11,BC=21 4F 5A DD 21 31 5A CD 8A 42 F1
. Eop
=====
. SYS7F/FIX - Patch to LDOS 5.3 MAX-80 SYS7/SYS - 06/29/87
. Corrects fixup on "SYSTEM (SYSRES=n" with bad "n"
. Apply via, PATCH SYS7/SYS.SYSTEM SYS7FM
D11,8B=2F:F11,8B=2E:D11,93=27:F11,93=26:D11,97=23:F11,97=22
D11,B6=F1 21 49 5A DD 21 2B 5A CD 8A 42:F11,B6=21 49 5A DD 21 2B 5A CD 8A 42 F1
. Eop
=====
. SYS7G/FIX - Patch to LDOS 5.3 SYSTEM (SYSGEN) command
. Corrects sysgen of SVC table interface after SYSTEM (SVC)
. Apply via, PATCH SYS7/SYS.SYSTEM SYS7G
D0A,4B=CE:F0A,4B=CD
. Eop
=====
.TOD01/FIX - Patch to TOD0/APP of PRO-WAM 2.X
. Keeps TOD0 from eliminating things on certain ADDs
. Apply via PATCH WAM1/APL TOD01
D5E,E1=F7;F5E,E1=D5
.Eop
=====
. WAM21/FIX - Patch to PRO-WAM 2.01a
. This patch improves PRO-WAM 2.01a so that you can activate PRO-WAM
. from within a JCL file via a //KEYIN macro. Warning: Do not attempt
. to issue another DO command. Invoking anything which uses the DOS
. @KEYIN SVC will fetch its input from the JCL file - not too useful!!!
. Apply patch via, PATCH PROWAM WAM21
D03,8E=64;F03,8E=45;D03,B3=64;F03,B3=45;D03,B6=10 0B;F03,B6=F1 0A
D07,DF=62;F07,DF=61;D0A,DA=CD 53 47;F0A,DA=32 73 3C;D0B,70=CD 46 47;F0B,70=32 1B 3D
D14,16=2A 09 3F 7E E6 8F 32 37 45 32 1B 3D C9
F14,16=00 00 00 00 00 00 00 00 00 00 00 00 00
D14,23=32 73 3C 3A 37 45 E6 0F FE 0D C0 11 C0 00 3E 45 EF C9
F14,23=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D16,3E=47 47 4D 47 50 47 54 47 57 47;F16,3E=00 00 00 00 00 00 00 00
. Eop
=====
. SYS0AFLG/FIX - Patch to LDOS 5.3 SYS0/SYS - 07/04/87
. This patch initializes AFLAG$ to the default value of 1.
. Apply using "PATCH SYS0/SYS.SYSTEM USING SYS0AFLG"
D05,C1=01:F05,C1=00
. EOP
. SYS8AFLG/FIX - Patch to LDOS 5.3 SYS8/SYS - 07/04/87
. This patch inserts the code to retrieve the value in AFLAG$.
. Apply using "PATCH SYS8/SYS.SYSTEM USING SYS8AFLG"
D00,F5=F5 3A 67 47 6F F1:F00,F5=2E 01 00 00 00 00
. EOP
```


SYSTEM
SYSTEM
KBDSCN
KBDSCN
DRIVER
DRIVER
PSTATU
PSTATU
C STARS
BØØT