NORTHERN BYTES

Volume 6 Number 6

GREETINGS! Welcome to another issue of NORTHERN BYTES. I hope you all had an enjoyable summer!

In the last issue I published a Compuserve number for The Alternate Source. Unfortunately, that was the starter kit number, and we didn't know that CIS would insist on changing it when we got a "real" account. So, the REAL Compuserve number is 72167,161. TAS also has an account on DELPHI, the user ID there is TASIO (The Alternate Source Information Outlet). Again, I want to emphasize that these accounts are checked very infrequently, so you should not use them to send time-critical messages.

The high cost of long-distance telecommunicating continues to tumble, at least for those of you who do not live back in the boondocks. See the article elsewhere in this issue on GTE Telenet's PC Pursuit. I might also mention here that if you have thought about signing up for MCI Long Distance Telephone Service but just have never got around to doing it, I know of a way that you can get it with NO installation charge (and, of course, NO minimum monthly charge). You probably know that MCI is less expensive than AT&T for long distance calls within the U.S. and Canada, but did you know that they are a lot less expensive than AT&T on most international calls? You may not even have realized that you could dial international calls using MCI, but you can (to many overseas points), and you can save a lot of money by doing so. Drop me a line and I'll see that you get a form that entitles you to sign up with MCI for no service charge.

Turning to other things, the long-awaited Alan Johnstone modifications to NEWDOS/80 are finally available from The Alternate Source, on TAS Public Domain Library Disk #ND-1. If you use NEWDOS/80, you will definitely want a copy of these!

LETTERS DEPARTMENT

Reminder: Persons sending letters intended for publication should send them on magnetic media or via MCI Mail (especially if longer than a couple of paragraphs). If you are NOT using Allwrite (or Newscript) and your word processor offers the option to save your file in ASCII format, please do so (especially if using SuperScripsit!). Your cooperation in this matter will help us to bring you a better newsletter!

Dear Jack,

Just received Volume 6 Number 4 and glad to see you're still hanging in there. With the pace of things in my life I don't get a chance to read everything I want to or write letters to comment on things that I read. But I'm home sick for a couple of days now so I'm trying to catch up a little.

We (your newsletter gets passed around here) hope you continue NORTHERN BYTES although I'm afraid we are as guilty as anyone about not sending you money to continue. Those of us who get the most out of your newsletter and opinions are also those of us who've been around the longest and have seen many things we've liked just go away no matter how much we supported it. The last publication I liked well enough to send in a three year renewal subscription for was 80-US which ceased publication just 60 days later. I don't want CREATIVE COMPUTING as a replacement or I would have subscribed to it in the first place. But I don't have time for writing and hassling so I just vowed I would never do such a thing again.

In fact, every time I try to encourage someone to continue something as well as they have been doing it, my actions backfire. You've heard of people causing rain by washing their cars? When you need a lot of rain that just isn't enough. In 1982 I was able to save southeastern Virginia from the worst drought of the century by tipping my paper boy for the first time in three years.

For 36 months he delivered my paper on time, placed inside my screened front door and wrapped in plastic despite the lack of rain. In a gesture of benevolence that I had resisted since his predecessor thanked me for a tip with a display of contempt, I began including a dollar tip with each collection. Immediately, he interpreted this as a sign of weakness from a "fish" and must have decided he was somehow graduated from small time paper boy to big time publication distributor executive.

Obviously, he had to respond to the stimulus. From the first "tip-day" on, he demonstrated his new-found status by refusing to wrap the paper in plastic. Apparently, the extra effort that takes is above that which should be expected from a distribution executive. Next, with the windfall profit from the tips, he hired a weather consultant and an old Indian rain-dance specialist. In a remarkable concert of coordination, he was able to tell exactly how late to be with my unwrapped paper so that delivery could be coincided with a daily downpour that eventually ended the drought.

But it took extraordinary measures. Almost daily he discovered inventive and creative new ways to increase the time it took me to retrieve my paper. From careful placement inside the screened door to a casual toss into the bushes did not initially allow enough time for it to rain. For a real drought, you need to discover the roof. And he did!

So, dear Jack, I would not want anything to happen to you and NORTHERN BYTES so I'm really doing you and all of us a favor by NOT sending you any money for future issues. What the enclosed $10 represents is thanks for the last five issues -- NOT the next five. If you have to cease production, don't blame it on me. Remember, the money isn't TIPS (To Insure Proper Service) in the future!

Now on to a different subject -- Disk Operating Systems. Somewhere in this issue you said someone should write a book about the rise and fall of the TRS-80 including a chapter on NORTHERN BYTES. Well, I plan on doing that when I retire in about 30 years. And you can be sure you will be included because you and I go way back. You were one of the 6,000+ owners who received (and contributed to) our TCS newsletter which was produced on a cassette-based Model I computer!

I've got some real sea stories to tell in that book! I'm still using that same computer (which now has 10,267 hours on it according to the elapsed time meter I installed as my first modification) to write this letter. (Seems like it should be hundreds of thousands of hours but meters don't exaggerate like estimates do.) I bought it in September of 1977 at the grand opening of Radio Shack's first Computer Center in the ground floor of the Tandy Center in my hometown of Fort Worth. I just happened to be back there at the time to help my parents move into a new apartment.

And I don't just keep it around as a souvenir. As you know, in 1982 I opened a floppy disk drive service shop and retail store specializing in disk-related products. Along with an Apple II, an Atari, three Model III's and another Model I, this Model I continues to be my workhorse for testing drives after they come off one of the benches. And my wife uses it with my own self-written relational software to do all of our bookkeeping, invoicing, inventory, payroll, taxes and forms-creation on a ProWriter 8510.

That's why I wanted to mention my DOS experiences. Maybe it's because I write my own software and, whether you can call it sophisticated or not, you would at least have to call it intricated! And the only DOS it works flawlessly with is NEWDOS/80 Version 2. But coming to that conclusion was anything but simple!

Back in the single-density days, I used TRSDOS on a friend's computer just long enough to decide that I would stick with cassettes which were more reliable. But I eventually broke down and bought a drive for myself when NEWDOS+ came with it. I never had a problem with NEWDOS+ except it was only single density.

When I bought a Percom Doubler, I discovered I had to use their DOUBLE DOS because there was no way to modify NEWDOS+. Then, when I upgraded Electric Pencil to Version II, I discovered that it would handle any double density DOS except DOUBLE DOS. DOSPLUS was becoming popular by then and I was tasked to write an inventory program on a system that used it so I took everybody's word that it was good. What a disaster! I'm just glad I had the opportunity to discover how dangerous it was before I went out and bought it for myself!

With DOSPLUS (all three versions I tried), I kept losing parts of files, line numbers changed and parts of BASIC programs began to behave as though the commands meant something completely different. To this day, it is the only DOS I've ever encountered that not only fails to backup a disk sometimes but destroys the original in the process!!! Calls to Micro Systems Software were probably interpreted as claims for the $100 reward they were offering at the time to anyone who could prove their system had a bug because all they would say is that I probably had a bad computer or disk drive.

They were right about one thing — all DOSPLUS errors were associated with disk I/O. In fact, that's how I got into the disk drive service business — trying to find something wrong with my drives! It was a long time, however, before I could prove to myself that the problem really was with DOSPLUS. I eventually wound up with a simple benchmark program. It consisted of a single line BASIC program with a single command:

```
1 LIST
```

If you would like to try it on DOSPLUS, the only thing you have to do is simulate the same conditions I needed for the inventory program I had written for the U.S. Navy. That is, you have to invoke AUTO in DOS so that when the reset button is pressed it will automatically load BASIC to use variable length files and reserve space for seven buffers so that as many as seven data files can be opened at one time. Now, all you have to do is press the reset button and if everything is successful, the program will automatically load and run which does nothing more than list itself.

But you will find that it doesn't always do that if you try it enough times. Usually it only takes about 20 tries before something unexpected happens. But sometimes it happens on the first or second try and sometimes it takes as many as 40 or 50 tries (or hundreds if you use fewer file buffers). But eventually it will not work. Sometimes it just goes to "silent death" — everything hangs up and nothing happens. Sometimes the "Memory Size?" question of Level II BASIC comes up. Sometimes everything seems to work except the benchmark program just doesn't work. When you LIST it the line number 1 has been changed to line number 250 or some other number. And sometimes the program seems to work and does list itself. But when you try to LIST it from the keyboard, there is nothing there to list. It's as though the program was deleted with NEW. Yet you can type RUN and it will still run and list itself! Try it yourself.

If the same thing always happened consistently we might have been able to figure it out. But with so many different symptoms from a simple single-line, single command program leading us in so many directions, me and my friends wasted countless thousands of man-hours on wild goose chases. And those same symptoms showing up intermittently in my complex inventory data base program used by unexperienced operators was just too much for the Supply Department who had the entire computer taken out of the building and turned in as excess!

I've tried the benchmark program on dozens of different TRS-80 computers with different board versions and all the different ROM versions, different brands of doublers, different kinds of disk drives and on different days using different diskettes with different versions of DOSPLUS. The only common denominator is TRS-80 Model I and DOSPLUS!

The frustrating part for me is the fact that even though most of my friends and customers who used a Model I with DOSPLUS will admit my benchmark program does fail as I've described, they said they'd had no problems with their own programs so they would continue to use it. However, I suspect they were experiencing problems and didn't even realize it. It's easy to assume you probably hit the wrong key when things are happening fast and something strange happens once but not the second time you try it. I sometimes wonder how many times people attributed DOSPLUS errors to that, glitches on the power line, operation during a lightning storm or just plain coincidence. There is no way to ever know.

I find it very interesting, however, that many DOSPLUS users brought me their disk drives to service because of "errors" they had been getting. But often I could find nothing wrong enough to have resulted in errors. I think it is significant and not coincidental that all the DOSPLUS errors I have encountered can be traced back to disk I/O. And it's also interesting to note now that most of my customers previously using DOSPLUS now use LDOS (because it is more "professional" in their opinion). And, conveniently enough, most of them either installed surge suppressors or went to Model III, 4 or 4P at the same time to account for disappearance of their errors rather than admit it could have been due to DOSPLUS.

After losing about a year of fighting the DOSPLUS battle, one of my friends asked me to evaluate MULTIDOS by trying to implement my infamous inventory program on it. I did and it was the first DOS that I could use to prove DOSPLUS was definitely defective and that I wasn't necessarily losing my mind after all. However, MULTIDOS has one problem that prevents me from using it.

Because I got double-sided drives from the very beginning, the only way I could use both sides was to modify the drive so that each side was assigned a different drive number. There was no DOS for the TRS-80 that would support a drive with two heads back in 1979. I still use the concept today because my double-sided 80-track drive can handle up to 444 separate files on a single diskette (using NEWDOS/80's ability to assign six granules to each directory so I can get 222 files on each side). I need that capacity due to the structure of my customer account status programs. The deficiencies of conventional information accessing and manipulation techniques are overcome by using individual short files rather than one long one.

None of the other DOS's have a problem with each side of a diskette having different drive number assignments. But when MULTIDOS opens a file on one drive and then retrieves data from another file on another drive to transfer into the first one, it assumes the head on the first drive has not moved since it did not command it to move. That's acceptable as long as the two drives are physically different. But the way I have my drives configured with the different drive numbers assigned to opposite heads of the same drive, you can imagine what happens. When the DOS commands the "second" drive to move to the appropriate track, the head on the "first" drive is dragged along for the ride because they are screwed together. Then when data is to be written into the open file on the "first" drive, MULTIDOS just starts writing over data on whatever track it happens to be on. Apparently, all other DOS's do some sort of track number checking or something before blindly charging off into a writing mode.

By early 1982, I was beginning to sour on the prospect of spending hundreds of dollars for a Disk Operating System sight unseen because of the bad experiences I had encountered. I had not wasted any money on bad DOS's up to that point because TRSDOS came with the manual I bought, NEWDOS+ came with the drive I bought and DOUBLEDOS came with the doubler I bought. I was fortunate to have learned about the dangers of DOSPLUS and MULTIDOS for nothing by trying to write a program for someone else's systems using them. But I still needed a DOS for myself that worked! And by then it was obvious to me that piracy was the only means left by software developers for anyone to evaluate software unless they were rich enough to afford an IBM computer.

Yes, I truly believe software piracy is almost entirely the fault of software developers themselves and I don't blame anyone for doing it! It is the result of all the poor and (not or) overpriced software which is just about the only choice the buying public has been offered! Ah, but that's another series of stories for another time...

I was saved from having to pirate a reliable DOS by a friend in early 1982 who gave up computing altogether. He sold his computer and gave me all of his disks including the original disk of NEWDOS/80 Version 2 and the original manual, both of which he had bought legitimately and was the registered owner of. So, even though I didn't buy it, I didn't have to pirate a copy either.

I have briefly tried to use LDOS at the insistence of a friend who believes it is the only "real" DOS for the TRS-80, but I almost lost him as a friend one night over a difference of opinion about it. He reflects the same "attitude" about LDOS as you describe in your newsletter. I found LDOS so awkward and cumbersome to use I just didn't have the time or patience to learn all the idiosyncrasies. To me, a computer is supposed to be easy to use. I don't particularly care how many unused and unneeded features something has just for the sake of having them if it's to the detriment of usability by simple-minded persons like myself. Enough said!

As it turned out, NEWDOS/80 Version 2 works, is extremely simple and will reliably do much, much more than anything I would ever want a single DOS to do. In the three years I've used it I have had no problem whatsoever. And I don't have to carry a manual around with me wherever I go.

Well, Jack, good luck to you. I'll write again the next time I get sick. Business is so good that'll probably be the only chance I get. A fellow called from Pennsylvania the other day and said he heard about me from you. You must be referring a lot of people to me because I'm getting drives from all over the country now and I'm not advertising anywhere.

Our rates are still the same as when we started — $25 for single-head drive cleaning, lubrication and alignment ($5 more for any dual head drive) and a flat $25 more if the drive is one of the 20% that needs "troubleshooting and repair" in order to achieve proper alignment. We still include all parts except heads and motors and still give the option of returning it at no charge if that's the problem. The only thing we do different now is for commercial accounts who are continuous customers. Instead of a discount we charge them a flat $35 per drive (plus cost of heads and motors), single or dual headed, whether a repair is necessary or not.

I hope NORTHERN BYTES continues and say hello to Charley Butler for me the next time you talk to him.

Sincerely,

Les Logan, 1215 Norview Avenue, Norfolk, Virginia 23513
[Les' business telephone number is (804) 855-6206]

[As mentioned above, Les Logan was one of the founders of TCS, which was probably the closest thing to a national user group that TRS-80 users have ever had. In fact, it's worth noting that one of the first pieces of national advertising for The Alternate Source was an insert in the TCS newsletter. I know, because that's how I first heard about TAS!

I've been sending Les copies of Northern Bytes on a complimentary basis. Partly that grew out of the newsletter exchange we used to have with TCS - when TCS was disbanded, I just kept sending the newsletters to Les. But mostly it's been sort of an attempt to repay Les for all the GREAT information that he used to put into the TCS newsletter. It's really too bad the group folded, because the TCS newsletter was one of the very best sources of information for early Model I hackers. I know I got a lot of benefit from the group, probably much more than I ever put into it at the time.

So, as far as I'm concerned, Les deserves a lifetime complementary subscription to NORTHERN BYTES, because if there had been no TCS newsletter, it would have probably taken me a extra year or so to learn what I now know about the TRS-80. The publication was that valuable to us early users! Les, I'm considering your generous donation as just that - a donation to keep NORTHERN BYTES going. I don't feel that you owe me a dime!

Regarding DOSPLUS, I have never done any real programming on it, so I have never experienced the problems mentioned. Why don't I use DOSPLUS? Mainly because it is much too sensitive when formatting disks! I could format a disk using DOSPLUS 3.5 and have it lock out four or five tracks, then turn around and format the same disk using NEWDOS/80 (or any other DOS) and there would be NO tracks locked out (nor any other problems). I just didn't care for having tracks locked out on perfectly good disks!

My gut feeling (although I have never disassembled DOSPLUS to prove this) is that perhaps DOSPLUS makes ONE attempt to format and verify a track, and if that doesn't work, it locks out the entire track. All other DOSes will at least make a few attempts to format a track before giving up. The problem is that a track may not format on the first attempt for any number of reasons that have nothing to do with the quality of the diskette, such as power line glitches, momentary speed variations in the drive, or other momentary faults that would not be present during the second attempt. Again, I can't prove this, but it seems as reasonable an explanation as any.

I did get a patch from Micro Systems Software to make DOSPLUS use a less sensitive formatting pattern and that did help a little, though it didn't solve the problem completely. Worse yet, with that patch applied I found that occasionally a disk that really was of marginal quality would make it through the formatting process.

The formatting problem alone was enough to dissuade me from further attempts to make any serious use of DOSPLUS. Which in a way is too bad, because I think that DOSPLUS 3.5 has some really nice features.

However, the problem reported in Les' letter is one I've never heard about before. If other NORTHERN BYTES readers are having similar problems with DOSPLUS (or better yet, if anyone has found a solution to those problems), please drop us a line and let us know the specifics!]

Dear Jack,

I am so mad I could just... That's the trouble, there doesn't seem to be anything TO do. I am talking about mass distribution of pirated software by school district officials. How bad is it? It is so bad that... Well, let me tell you about it.

By day, I am a technician in a large school district media center. There, a high speed cassette copier is kept busy most of the time. Sometimes an administrator will sit at it for hours, running off a shopping bag full of cassettes. More often a student co-op is sent to do the actual copying.

I have argued with school district administrators for years. I say that buying one copy of something and making hundreds of copies is wrong. The administrators insist that, since they don't sell the copies for profit, there is no law against it. Besides, there is nothing that the copyright holders can do about it.

Nights and weekends, I write computer software. One administrator, after previewing one of my proframs, said "It's very good but I can't CSAVE it. When you get that fixed, I'd like to look at it again."

Sure thing mister. Would you like to borrow my check book for a few days too?

The school district offers a SCRIPSIT workshop. Everyone who takes the workshop, naturally, gets a copy of the program. There must be close to a thousand copies of SCRIPSIT in my district alone. I have only seen one in the original Radio Shack binder.

But that's not the worst of it. I just learned that there is a well organized, nation wide, Computer Consortium. It is huge and it is supported by our tax dollars. A two hundred page catalog is full of popular software packages, all at a single price. Fifteen dollars. That includes a Sentinel diskette, photo-copied documentation and a nice plastic binder.

I wouldn't swear that all Consortium programs are pirated, but I don't see how they could be anything else. Add it up. Ten bucks for the binder, two for the disk and three for printing. That doesn't leave much for royalty payments does it? The Consortium does not allow previewing, so the media center orders preview materials from the "real" distributor. Then, if they decide to buy, they buy it from the Consortium. That alone seems dishonest to my way of thinking.

A program that YOU wrote, might be in every classroom in the country right now. You should be hanging a gold plated floppy over the fireplace in your new mansion. But you're not because you only got paid for one copy. You have no way of knowing. That is how bad it is.

Robert F. Noonon

[The above two letters offer two views of software piracy. I agree that the person who gets stung buying a piece of lousy, overpriced software is probably going to be fairly receptive to the next offer of a pirated copy of a program. But, folks, if you do get a pirated copy of something and find that you really do like like the program and that you actually use it more than once or twice, you really should buy a legitimate copy (if possible - that's another story. I know of one producer of TRS-80 software that sells their products by mail order only (that is, they do not sell through dealers or distributors) but refuses to ship their products to Australia (and probably other countries outside the U.S.A. as well). Naturally, that company's products are freely available there via the "pirate grapevine". Many of the users in Australia would prefer to own legitimate copies of these products but can't get them, so what are they supposed to do? Chalk this up as a tiny bit of insanity within an otherwise fine company).

However, Mr. Noonon addresses an issue that cuts to the very heart of our society. I have heard it time and again in many and various ways, and this is but one symptom. The problem is that many school officials somehow consider themselves part of the "intellectual elite" and therefore above the law! I'm serious - I think that many people in the educational profession have somehow gotten the idea in their heads that the laws are made to be used for their benefit, and when a law somehow inconveniences them or threatens to impede their goal of "educating" our youth (which, you may have noticed, they've been doing a lousy job of lately), they think they can ignore the law with impunity.

(If you find this hard to believe, may I suggest one book that might open your eyes to what is really happening in some of our schools nowadays. It's called Child Abuse in the Classroom, edited by Phyllis Schlafly, and published by Pere Marquette Press, Alton, Illinois 62002 and by Crossway Books, Westchester, Illinois 60153. This book consists of "selected excerpts from the Official Transcript of Proceedings before the United States Department of Education in the matter of the Proposed Regulations to Implement the Protection of Pupil Rights Amendment ... also known as the Hatch Amendment." In other words, transcripts of testimony by parents, public school teachers, and interested citizens describing accounts of psychological abuse of children in the public schools. It has nothing to do with computers, but it sure does reveal the attitudes of some public school officials!)

Think about this a moment - here a school is handing a student a bag of cassettes and a master copy of a program and saying "run us off a few copies." Folks, not only is the school system participating in a criminal act, they are getting a minor student (who is in their care and control) to aid and abet them in this act. <u>The person who induces a student to do this should go to jail</u>, even if it's the Superintendent of Schools himself, and I don't mean for just a weekend lockup, either.

It is a well know fact in the software industry that school systems are the <u>worst</u> offenders insofar as software piracy is concerned. I'm not talking about students swapping programs with other students (which is largely beyond the control of the school anyway). I'm talking about a district that buys one copy of a program, and suddenly every computer user in the district (be they administrator, teacher, or student) has a copy. And it is a difficult problem, because who wants to sue a school district (it creates a bad image among the folks of the community, who by and large don't really understand the problem).

What can be done? Well, you're reading this publication, which implies that you know something (probably a great deal) about computers and about the software piracy issue. If you have school-age children, <u>ask</u> them if they know of any illegal copying of programs in their schools. If you discover this sort of thing going on, why not go to your next school board meeting and ask why Federal copyright laws are being violated, and what kind of example do they think this sets for the students (this works particularly well if the press is in attendance, especially if you have taken the trouble to brief the press on the issues involved - this could be done with a one or two page handout. After all, the school officials probably won't listen to you alone, since you're just a lowly <u>parent</u> and therefore not one of the intelligentsia).

But, I don't realistically expect to see this wholesale piracy stopped until some software manufacturer actually sues a teacher, principal, or some other school official for about two million clams. Okay, which of you Big Software Manufacturers wants to wear the black hat for this act? (Don't everybody jump at once, now).

Final comment - my remarks above are by no means intended to apply to the many honest and decent educators in our country. And, I do understand that school budgets are always tight, but better to do without a computer altogether than to teach students that stealing is okay. If you are an educator that has practiced this form of theft in the past, please destroy your illegal copies now and refrain from doing this in the future. And above all, don't ask students to do your dirty work for you. Remember these words from The Living Bible (The Book): "'There will always be temptations to sin,' Jesus said one day to his disciples, 'but woe to the man who does the tempting. If he were thrown into the sea with a huge rock tied to his neck, he would be far better off than facing the punishment in store for those who harm these little children's souls. I am warning you!'" (Luke 17:1-2)]

Dear Jack,

Having just obtained another copy of Northern Bytes from a friend, I am reminded that "The Road To Hell is Paved With Good Intentions"! By this I mean I have intended for some time to send you my Visa number to help support your efforts to keep that marvelous little "Rag" with its' wealth of information coming our way.

With this in mind I am submitting my Visa number to cover future issues. Also I am including a check to cover some of the bootleg back issues I have in my files (some were obtained through The Alternate Source). While I know this is not necessary, your publication and efforts are really appreciated by those of us who enjoy tinkering (the word "hacking" has been misrepresented by the press).

On another subject -- it is not necessarily true that the LNW type doubler requires the TI flag to be set to E, as the Apparat instruction manual would seem to indicate. As a matter of fact, I find that the only result I got from that setting was that my disks were no longer compatible with the Percom doubler which I also have. The LNW doubler works just as well with the TI flag set to C or CK, same as the Percom doubler.

Floyd C. Trimble

[Occasionally I get a letter like this, and always appreciate the encouragement and support of our readers. I just wish there were a lot more of you out there, because it would make it a lot easier to justify the time and effort I put into each issue of NORTHERN BYTES. Thanks, Floyd, your support is very much appreciated (and thanks also to those other readers who have also contributed to NORTHERN BYTES "above and beyond" what we have requested).

I refuse to give up the use of the word "hacker" in its original sense. I am a hacker and proud of it, and blast it, I am not about give up the use of the term because ABC, CBS, NBC, or anyone els misuses it. If certain groups can force the media to use silly terms like "chairperson" or "fireperson" (presumably because they are supposedly somehow derogatory to women - note, however, that the term "woman" itself contains the "man" ending! I'm surprised they haven't tried to do something about that as well!), then why should we computer hackers allow the press to give us a bad name. TO CALL SOMEONE A HACKER IS A COMPLIMENT! When the press says that a "computer hacker" broke into a computer system to steal or vandalize data, what they are really saying is that an intelligent person who likes to find out how his computer works got carried away and crossed the line into an illegal activity - in other words, they are paying the person a compliment while reporting his crime. That may not be their intention, but when the term "hacker" is used in that type of story, that is how it comes out (at least to those of us who know the origin of the term). Perhaps we need a "Computer Hacker's Anti-Defamation League" to "educate" the news media (and the general public) as to the proper use of the term "hacker".

Regarding the NEWDOS/80 TI flag, I also have found that the "C" specification works with an LNW doubler. I have also found that both "C" and "E" work with the Holmes Engineering doubler. I normally use the "C" specification on all my Model I format disks. Does anyone know of any doubler (other than the Radio Shack doubler) that will <u>not</u> work with the "C" specification?]

Dear Jack,

I have an Anadex M-8000 printer for which I need a schematic and pin-out. It was also listed at one time as the TRIAD Model 3070; the main board has a number of 1700-5263-04. I would like to obtain the complete manual, but that is probably too much to wish for.

Andrew Johnson, P.O. Box 3689, Warner Robins, Georgia 31099

[Readers, can anyone help with this one?]

Dear Jack,

... If you own a Model 4 and are using the Model III version of Superscripsit, be careful. Several times I have been using TRSDOS 6.2 and then decided to put Superscripsit in. Instead of turning off the computer first, I just pushed the reset button. Superscripsit then has a bad habit of locking up (usually when I am about finished) and the only way I have been able to regain control and start over without the same problem is to remove the disk and turn the computer off and back on, thus losing whatever I was typing.

A question I would like answered in Northern Bytes, because I'm sure I'm not the only one with it, is why does it seem as if almost all of the articles deal with NEWDOS/80 and there are not more dealing with TRSDOS or LDOS? I really liked the article about the TRSDOS 6.2 patches and plan to try some of them out very soon.

Keep up the good work.

Sincerely, Charles E. Tatman

[First of all, if I were you I'd make it a priority to learn how to recover your text in memory before turning off the machine. Even though the computer "locks up", your text is still in memory (until you cut the power) and (in theory, anyway) you should be able to examine it using DEBUG and then perhaps to DUMP that segment of memory to a disk file. The whole recovery process would be much easier if you were using another word processor, for two reasons: First of all, Superscripsit has the screwiest format for text storage (on disk, anyway) that I have ever seen, and that has to complicate the recovery process. And second, some word processors have a method to automatically recover text in memory after a system crash. For example, with Allwrite, you simply type AL * from DOS READY and Allwrite will attempt to restore your text, so that you can save it to disk (most other non-Radio Shack word processing programs have a similar feature). Anyway, maybe some othe Superscripsit user will write in with some more help for your specific problem.

As for your other question, the bias toward NEWDOS/80 is not by design. If more authors would contribute TRSDOS 6-related articles, I'd run more of them (I can't publish what I don't have). I suspect that you will see more and more coverage of TRSDOS 6 in future issues, as more users become comfortable with it. Also, it would help a whole lot if Logical Systems, Inc. would drop the price on their TRSDOS 6 source code listings to a more reasonable price - $99 per book (and there are three books!) is no problem for any large company that might want to "rip off" parts of the code for whatever reason, but it's certainly prohibitive for the average hacker (and that includes your editor!). Granted, Apparat was never very free with technical information on NEWDOS/80, but then it's been around a lot longer than TRSDOS 6, so there is more knowledge about it in general circulation.

As for LDOS, if you enjoy running LDOS on a Model I, you might also enjoy flogging yourself with whips and chains. My attempts to use LDOS a couple of years ago were met with nothing short of extreme frustration. Possibly one of the worst things about LDOS is what I call the "sudden death" mode, where you do something that LDOS doesn't like and it just up and dies on you, necessitating a reboot (this actually appears to happen rather randomly). Unfortunately, this "feature" has been reported to me by other users of LDOS, and even occasionally by TRSDOS 6 users. No, I'm not saying that I won't run LDOS-related articles (and I try not to discriminate against them), but I don't experience feelings of extreme joy when I get one, either!]

Date: Tue Aug 27, 1985 2:29 pm EDT **RECEIPT
From: Bob Seaborn / MCI ID: 268-7906

TO: *Jack Decker / MCI ID: 102-7413

Jack, I just received Volume 6 Number 4 today. Much thanks for getting it to me. I'm just sending this with a couple of comments.

First, the rewrite of the NEWDOS/80 Disk Basic Ampersand (&) Routine that appears on page 22 has the same listing for both the "before" and "after". They're both for the "before", so it might be an idea to get the "after" published soon. I'm sure that I'm not the first to notice this, so if I'm repeating what others have already told you, please accept my apologies for wasting your time.

Next, with regards to the zap for SUPERZAP/CMD provided by Paul Fransen on page 14, I have taken the liberty of changing the 5BH byte to 09H and the two 0AH bytes to 08H. This allows the use of the left and right arrows to scroll through the sectors rather than the up and down arrows. I'm too used to Super Utility which uses the up/down arrows to change tracks and this will allow more compatibility.

I'll be in touch again.

Bob

[Actually, Bob, you WERE the first to notice the bug in the Ampersand function article. Thanks for pointing it out, and you'll find the corrected article elsewhere in this issue.]

Dear Jack,

We of the Tandy Hobart Users' Group (THUG!) wish to express our sincere thanks for sending us Volume 6, Number 1 "Northern Bytes". I (in particular) was fascinated when reading some of the articles! As we are a small group of some 25 people, we have no newsletter of our own, but possibly you might find the following useful and, if so, publish it.

Your readers are undoubtedly aware of the date bug in Radio Shack's much maligned CP/M 3.0+. There are actually two date bugs. The first is responsible for adding one day onto the date entered after booting the disk on power-up. This was the subject of an excellent article by Charles Alexander in "80 Micro" (October 1984, pp.114-119). The second bug is (more?) serious as October 1 becomes January 18, October 2, January 19, and so on. After discovering this the hard way, I wrote to Mr. Alexander. The second problem arises because the RS programmer used the A register in which to accumulate the total. However, as this is a 256 byte register, there are insufficient bytes to hold the 365 days of the year. On a separate sheet I have included the patches Mr. Alexander sent me to correct these faults. They work!

Yours sincerely, Herbert Smith
G. P. O. Box 1271 N. Hobart, Tasmania 7001, Australia

[Thanks, Herbert! Here are the patches that Herb sent!]

BIOS PATCH TO CORRECT ADDITION OF MONTHS

NOTE! In both versions, you must first do the following:

Put your working version of DOS in drive B:
Drive A: should contain SID and SET
Your default drive should be A:

```
SET B:CPM3.SYS(RW)
SID B:CPM3.SYS
```

BANKED VERSION	UNBANKED VERSION
S0553	S04C4
16	16
00	00
DD	DD
21	21
00	00
00	00
5E	5E
DD	DD
19	19
0D	0D
23	23
C2	C2
D9	4A
F0	E7
2A	2A
87	F8
F1	E7
CD	CD
68	D9
F1	E7
2A	2A
F4	F4
D2	CA
DD	DD
E5	E5
D1	D1
19	19
06	06
00	00
09	09
.	.
WB:CPM3.SYS,100,5CFF	WB:CPM3.SYS,100,3FFF

Date: Thu Jul 11, 1985 12:54 am EDT
From: Clifford Richards / MCI ID: 266-3283

TO: *Jack Decker / MCI ID: 102-7413
Subject: Bug Fix

Exterminator - Fix for the "TO" Bug on ND/80. Have renamed Tony Domigan. He is now known as 'PETE' (4P) Domigan.

This completely eliminates this bug. Take Care!

Cliff Richards

[You know, if I'd sat and thought about it for a thousand years, I don't think that solution would ever have occurred to me, yet it's so obvious. Maybe I need a little of that Australian sunshine to help stimulate my brain cells, so I can come up with such original solutions to my problems!]

REWRITE OF NEWDOS/80 DISK BASIC AMPERSAND (&) ROUTINE

by Gil Spencer VK2JK

[EDITOR'S NOTE: Does this article look familiar? If it does, it's because we first published it in NORTHERN BYTES Volume 6, Number 4. Trouble is, I must have been half asleep when I formatted this one for publication, because somehow or other I managed to use the same listing for both the "before" and "after" code. The listing I printed (twice) was the "before" version, so I'm reprinting the entire article here, with the correct "after" code in place this time. Bob Seaborn wins the "eagle eye" award for spotting this error!]

I never think in octal. It's hard enough to work in binary, hex, and decimal. It always seemed to me that the default for the '&' function should be hex, not octal. I finally dug out the source code (from Apparat's Disk BASIC) which I found in SYS20/SYS. My rewrite fits within the required extra space. Although a quantity of bytes are changed, this is because the code is "re-arranged" more than because it is "re-written".

First, here is the Disk BASIC (&) routine found in Apparat's NEWDOS/80 version 2.0 - specifically SYS20/SYS, addresses 54C5H-5503H. If you are using SUPERZAP, address 54C5H is found at FRS 2, byte D1H and address 5503H is at FRS 3, byte 13H. Note that the four bytes at FRS 3, bytes 06H-09H (which are 01 00 FA 54) are loader codes and must NOT be changed.

```
54C5               00100          ORG     54C5H
54C5  D7           00110          RST     10H
54C6  4F           00120          LD      C,A
54C7  110000       00130          LD      DE,0000H
54CA  79           00140 Q54CAH   LD      A,C
54CB  FE48         00150          CP      48H
54CD  2022         00160          JR      NZ,Q54F1H
54CF  D7           00170          RST     10H
54D0  EB           00180          EX      DE,HL
54D1  D630         00190          SUB     30H
54D3  FE0A         00200          CP      0AH
54D5  3808         00210          JR      C,Q54DFH
54D7  D611         00220          SUB     11H
54D9  FE06         00230          CP      06H
54DB  3022         00240          JR      NC,Q54FFH
54DD  C60A         00250          ADD     A,0AH
54DF  29           00260 Q54DFH   ADD     HL,HL
54E0  3807         00270          JR      C,Q54E9H
54E2  29           00280 Q54E2H   ADD     HL,HL
54E3  3804         00290          JR      C,Q54E9H
54E5  29           00300          ADD     HL,HL
54E6  3801         00310          JR      C,Q54E9H
54E8  29           00320          ADD     HL,HL
54E9  DAB207       00330 Q54E9H   JP      C,07B2H
54EC  85           00340          ADD     A,L
54ED  6F           00350          LD      L,A
54EE  EB           00360          EX      DE,HL
54EF  18D9         00370          JR      Q54CAH
54F1  0E4F         00380 Q54F1H   LD      C,4FH
54F3  B9           00390          CP      C
54F4  2801         00400          JR      Z,Q54F7H
54F6  2B           00410          DEC     HL
54F7  D7           00420 Q54F7H   RST     10H
54F8  EB           00430          EX      DE,HL
54F9  D630         00440          SUB     30H
54FB  FE08         00450          CP      08H
54FD  38E3         00460          JR      C,Q54E2H
54FF  CD9A0A       00470 Q54FFH   CALL    0A9AH
5502  EB           00480          EX      DE,HL
5503  C9           00490          RET
```

This is the rewrite of the Disk BASIC ampersand (&) routine. Now the octal argument must be specified by '&O'. Hex argument may be specified by '&H'. No suffix (i.e. '&') now defaults to hex rather than octal.

```
54C5               00100          ORG     54C5H
54C5  D7           00110          RST     10H
54C6  4F           00120          LD      C,A
54C7  110000       00130          LD      DE,0000H
54CA  79           00140 Q54CAH   LD      A,C
54CB  FE4F         00150          CP      4FH
54CD  2828         00160          JR      Z,Q54F7H
54CF  0E48         00170          LD      C,48H
54D1  B9           00180          CP      C
54D2  2801         00190          JR      Z,Q54D5H
54D4  2B           00200          DEC     HL
54D5  D7           00210 Q54D5H   RST     10H
54D6  EB           00220          EX      DE,HL
54D7  D630         00230          SUB     30H
54D9  FE0A         00240          CP      0AH
54DB  3808         00250          JR      C,Q54E5H
54DD  D611         00260          SUB     11H
54DF  FE06         00270          CP      06H
54E1  301C         00280          JR      NC,Q54FFH
54E3  C60A         00290          ADD     A,0AH
54E5  29           00300 Q54E5H   ADD     HL,HL
54E6  3807         00310          JR      C,Q54EFH
54E8  29           00320 Q54E8H   ADD     HL,HL
54E9  3804         00330          JR      C,Q54EFH
54EB  29           00340          ADD     HL,HL
54EC  3801         00350          JR      C,Q54EFH
54EE  29           00360          ADD     HL,HL
54EF  DAB207       00370 Q54EFH   JP      C,07B2H
54F2  85           00380          ADD     A,L
54F3  6F           00390          LD      L,A
54F4  EB           00400          EX      DE,HL
54F5  18D3         00410          JR      Q54CAH
54F7  D7           00420 Q54F7H   RST     10H
54F8  EB           00430          EX      DE,HL
54F9  D630         00440          SUB     30H
54FB  FE08         00450          CP      08H
54FD  38E9         00460          JR      C,Q54E8H
54FF  CD9A0A       00470 Q54FFH   CALL    0A9AH
5502  EB           00480          EX      DE,HL
5503  C9           00490          RET
```

ELECTRIC PENCIL ZAPS

by Brendon Thompson

[This article is a merger of two articles which originally appeared in Christchurch-80, the official magazine of the Christchurch 80 Users Group, P.O. Box 4118, Christchurch, New Zealand!]

Running an 80-track disk drive double-sided and in double density has some advantages (599 grans free) and some disadvantages. One of the big disadvantages is that the Electric Pencil will not display the directory. To see what's on the disk you have to get into Mini-DOS and look from there. An unnecessary complication in a carefully uncomplicated word-processor.

The reason for this problem is that in NEWDOS/80, PENCIL normally expects the directory to be starting on track 17, sector 0 (disk sector 170). The directory on an 80 track double-sided double density disk does not start on sector 170, and cannot be persuaded or bullied into living there without unusual and inhumane treatment, so PENCIL won't read it. The problem to date has been insoluble.

Well fear not. Salvation is at hand. Here is a zap which will persuade PENCIL to read your directory wherever it has gone to. It is a long one, but well worth it.

```
                PENCIL01/SYS,09,3C
Change                                C9 01 AC 00 3A
        27 44 FE 02 20 05 32 40 6E 18 0A 3A 03 52 FE 31
        28 03 01 14 01 ED 43 4A 6E ED 43 35 6E CD 93 64
        2A 30 52 22 43 6E AF 32 46 6E 32 3D 6E 13 1A B7
        28 14 FE 2A 20 03 32 3D 6E FE 34 38 F0 FE 30 38
        EC E6 03 32 46 6E 21
   to                                 C9 CD 93 64 21
        3D 6E AF 77 F6 30 32 FF 6A 13 1A B7 28 10 FE 2A
        20 01 77 FE 30 38 F2 FE 34 30 FE 32 FF 6A 21 F7
        6A 11 3E 52 CD 1C 44 2A 30 52 01 02 00 CD 24 44
        C2 01 62 3A 4A 52 91 32 0E 6B 18 0A 44 49 52 2F
        53 59 53 3A 30 03 21

PENCIL01/SYS,09,8C   change 11 40 6E CD to 11 3E 52 CD

PENCIL01/SYS,09,99   change 11 40 6E CD to 11 3E 52 CD
```

REMEMBER to perform this zap on a backup, and verify it is performing properly before you copy it to your working disk. The zap can be done with a utility such as SUPERZAP, or from BASIC with Alan Johnstone's ZAPPER program.

Here is a patch supplied by IJG, the distributors of Electric Pencil, which some of you may not have seen. It is described as a mandatory zap to prevent the occasional dropping of characters in text entry mode at the end of a line or the end of a screen.

While it does not prevent this problem completely, this zap certainly cuts down the number of characters lost.

```
PENCIL/CMD,07,C2
Change  0D FA D7 55 1A
    to  0D C3 B6 60 1A

PENCIL/CMD,14,70
Change  00 00 00 00 00 00 00 00 00 00 00
    to  00 F2 3F 5A CD 73 5D C3 07 55 00
```

NOTE: Depending on what you have done with your PENCIL/CMD file before this, the relevant coding may have been shifted round in the file. In mine, the first was found at sector 7, byte 6B, and the second at 13, byte FD. Locate the code to change rather than zapping "blind", and try it on a backup first.

MODEL 4 SYSTEM STATUS FLAGS

(How to access the Model 4/4P system status flags using BASIC)

by Rowan Evans (phone 011-61-2-419-3164)

[This article is excerpted from the Mr. Evans' column, "THE PROPHET AND ORACLE SPEAK", which appears in the SYDTRUG NEWS, P.O. Box 297, Padstow, New South Wales 2211, AUSTRALIA.]

Note that the information presented here is for the Model 4 TRSDOS/LDOS 6.x ONLY! I strongly suggest that anyone contemplating using this information follow the machine code subroutine access method if portability is desired. I also cannot and will not guarantee the actual addresses to be valid for any release of TRSDOS/LDOS 6.x after 6.1.2.

Firstly, in TRSDOS/LDOS 6.x, there are 26 status flags, named A to Z - in the Misosys "Programmers Guide to TRSDOS/LDOS 6.x" they are labelled xFLAGS$ - where x is A to Z. The status flag table in 6.1.x resides at addresses 006AH to 0083H (page 0 in low memory). The function of some of these flags follows (not all are used and some that are are not very useful) -

CFLAG$ - System execution flags.
DFLAG$ - Spool, verify, type-ahead and MemDisk.
KFLAG$ - <BREAK> or <ENTER> pressed, CAPS lock and an indication of a character in the type-ahead buffer.
LFLAG$ - Step rate and number of sides questions on/off in Format and 8" question in FLOPPY/DCT.
MFLAG$ - Modem output port image.
OFLAG$ - Memory management port image.
SFLAG$ - System status flags.
VFLAG$ - Cursor controls and clock display.
WFLAG$ - Interrupt mask port image.

The useful flags of these are CFLAG$, DFLAG$, KFLAG$, LFLAG$, SFLAG$ and VFLAG$. The flag bits that are useful under BASIC are marked with an * next to them. Flag bits marked + are SYSGENable, so if they are modified under BASIC (or assembler or C or whatever) and then a SYSGEN is done, they will remain in effect. An explanation of what each bit in these flags is for is listed below -

<u>CFLAG$ - Address 0006CH</u>

Bit 0 - If 1, system will not allow alteration of HIGH$ via SVC-100 - useful if your program accesses system resources via the @CMNDR SVC, and you want to retain control of HIGH$.
Bit 1 - If 1, @CMNDR is executing. This is NOT a flag that can be modified by the programmer!
Bit 2 - If 1, then SYS1 is requesting keyboard input. If SYS1 is not resident when control is restored from your driver then the whole thing will hang messily!
Bit 3 - If 1, then the system is requesting execution via SET or SYSTEM (DRIVER= commands. Useful to see if your filter/driver is being activated in the proper manner.
+ Bit 4 - If 1, @CMNDR will ONLY execute system LIB commands (including RUN!!).
Bit 5 - Reserved by DOS.
+ Bit 6 - If 1, @ERROR will not output any messages. Useful if you want to handle errors in your own manner.
+ Bit 7 - If 1, @ERROR will send any error message to the buffer pointed to by the DE register pair. Useful if you want to modify any system error messages before displaying them.

<u>DFLAG$ - Address 0006DH</u>

Bit 0 - If 1, then the spooler is active.
+*Bit 1 - If 1, then type ahead is active. By setting this bit to 0, type ahead may be deactivated.
+ Bit 2 - If 1, then VERIFY (ON) has been set.
Bit 3 - Reserved by DOS.
Bit 4 - If 1, then MemDisk is active.
Bit 5 - Reserved by DOS.
Bit 6 - Reserved by DOS.
+*Bit 7 - If 1, then screen print will allow output of block graphics.

<u>KFLAG$ - Address 00074H</u>

*Bit 0 - If 1, then <BREAK> was pressed.
*Bit 1 - If 1, then <PAUSE> was pressed.
*Bit 2 - If 1, then <ENTER> was pressed.
Bit 3 - Reserved by DOS.
Bit 4 - Reserved by DOS.
+*Bit 5 - If 1, then CAPS lock is on. Set this bit to 0 to allow lower case input.
Bit 6 - Reserved by DOS.
*Bit 7 - If 1, a character is available in the type ahead buffer.

<u>LFLAG$ - Address 00075H</u>

+ Bit 0 - If 1, FORMAT will not prompt for the step rate of the drive.
Bit 1 - Reserved by DOS.
Bit 2 - Reserved by DOS.
Bit 3 - Reserved by DOS.
+ Bit 4 - If 1, FLOPPY/DCT will not ask the 8" drive question.
+ Bit 5 - If 1, FORMAT will not prompt for the number of sides to format the diskette to.
Bit 6 - Reserved by DOS.
Bit 7 - Reserved by DOS.

<u>SFLAG$ - Address 0007CH</u>

Bit 0 - If 1, then the system will only READ a file. The LRL will not be checked and the "file open bit" in the directory will not be set.

Bit 1 - This bit is set to 1 when an EXEC only file is loaded via @OPEN. To operate bit 2 MUST be set to 1!
Bit 2 - If 1, then an EXEC only file may be loaded. It must be set in conjunction with bit 1.
Bit 3 - If 1, then SYSTEM (FAST) is operational.
+*Bit 4 - If 1, then the <BREAK> key is disabled.
Bit 5 - If 1, then DO is in effect executing JCL.
+ Bit 6 - If 1, then extended error messages will be issued.
Bit 7 - If 1, then DEBUG will be turned on after execution of the executing program. If the executing program is EXEC only, the bit will not be modified, but debug will not be entered.

<u>VFLAG$ - Address 0007FH</u>

Bit 0 - These bits
Bit 1 - control the
Bit 2 - cursor blink
Bit 3 - rate.
+*Bit 4 - If 1, then the clock will be displayed.
+*Bit 5 - This bit toggles the cursor state (on/off??).
+*Bit 6 - If 1, then the cursor is non-blinking. If 0, the cursor is blinking.
Bit 7 - Used by the system to suppress blinking while it is executing code in the *DO driver, to prevent the blink task from changing state.

Now the easiest method to access/modify these bits, knowing their addresses, is as follows-

```
5 REM BASIC program to access/modify TRSDOS/LDOS 6.x system
10 REM  Copyright 1985  Rowan C. Evans
15 REM          Chatswood, N.S.W., Australia
20 REM flag bits.
25 BASE.ADDRESS=&H6A:REM Base address only for 6.1.x!!
30 INPUT"Which flag to examine ";FLAG$
35 IF FLAG$="" THEN END
40 FLAG=ASC(FLAG$):IF FLAG>90 THEN FLAG=FLAG-32
45 IF FLAG<65 OR FLAG>90 THEN 30
50 FLAG=FLAG-65
60 ADDRESS=BASE.ADDRESS+FLAG:VALUE=PEEK(ADDRESS)
70 GOSUB 500
130 INPUT "Modify this flag (Y/N) ";YESNO$
140 IF LEFT$(YESNO$,1)="N" OR LEFT$(YESNO$,1)="n" THEN 30
150 INPUT "Bit to modify ";BIT
160 IF BIT=-1 THEN 30
170 IF BIT<0 OR BIT>7 THEN 150
180 BIT=2^BIT:VALUE=(VALUE XOR BIT)
190 POKE ADDRESS, VALUE
195 GOSUB 500
200 GOTO 150
500 PRINT "Flag - ";CHR$(FLAG+65);FLAG$;" Current Value -
";VALUE
510 PRINT "Bits 7 6 5 4 3 2 1 0":START=128
520 FOR X=1 TO 8
530 IF (VALUE AND START)=0 THEN PRINT "0 "; ELSE PRINT "1 ";
540 START=START/2
550 NEXT X:PRINT:PRINT
560 RETURN
```

O.K., so now you can play silly buggers with the system flags in TRSDOS 6.1.x, what happens when you want to do the same sort of thing to a program to run under LDOS 6.x on a Model 12 or MAX-80 ?? These addresses aren't guaranteed, remember ??? Well, LSI was rather smart and put a Supervisor Call into LDOS 6.x to get the base address of the flag byte table. This SVC is @FLAG$, #101. Knowing this, we can now write a machine language subroutine for a portable version of our BASIC program. This machine language routine is -

```
        ORG     03000H   ; Anywhere, it's relocatable.
START   LD      A,101    ; Load A reg with SVC number.
        RST     28H      ; Do SVC
        PUSH    IY       ; Base address is in IY regs.
        POP     DE       ; Move it to DE regs.
        LD      (HL),E   ; Save low order byte.
        INC     HL       ; Next byte of calling variable.
        LD      (HL),D   ; Save high order byte.
        DEC     HL       ; Point to start of variable.
        RET              ; Return to BASIC program.
```

This routine must be called with an integer variable as its argument, otherwise the whole thing will BOMB!!!

So, the changes to make to the first program to accommodate the portability available under LDOS 6.x are -

<u>Add -</u>

```
5 I%=0:S$="":DATA 62,101,239,253,229,209,115,35,114,43,201
6 FOR X=1 TO 11:READ N:S$=S$+CHR$(N):NEXT X
7 DEF USR0=PEEK(VARPTR(S$)+1)+256*PEEK(VARPTR(S$)+2)
```

<u>Change -</u>

```
10 BASE.ADDRESS=USR0(I%)
```

And that's all! Have fun modifying your Model 4/4P's system flags!

MODEL I NEWDOS/80 "CLEAR" COMMAND ZAPS

by Bob Seaborn / MCI ID: 268-7906

I was very interested in Tony Domigan's zaps to the "CLEAR" command for the Model III (in Northern Bytes Volume 6, Number 5) - so much so that I duplicated them for the Model I.

These patches convert the "CLEAR" command to respect HIMEM and only clear the memory from 5200H to HIMEM on the DOS command "CLEAR" while the DOS command "CLEAR *" will clear all memory from 5200H to FFFFH and reset all user routing.

```
ZAP    SYS14/SYS,3,70 (I)
CHANGE FE 0D 28
TO     C3 75 51
ZAP    SYS14/SYS,4,88 (I)
CHANGE 00 00 00 00 00 00 00 00 00 00 00 00 00 00
       00 00 00 00 00 00 00 00 00 00
TO     FE 2A CA 79 50 FE 0D C2 76 50 2A 49 40 22
       BC 50 E5 D1 21 FF FF C3 94 50
END
```

[Editor's note: For those of you who missed them last issue, here's the Model III version of the same zaps, by Tony Domigan!]

```
ZAP    SYS14/SYS,3,82 (III)
CHANGE FE 0D 28
TO     C3 87 51
ZAP    SYS14/SYS,4,9B (III)
CHANGE 00 00 00 00 00 00 00 00 00 00 00 00 00 00
       00 00 00 00 00 00 00 00 00 00
TO     FE 2A CA 8E 50 FE 0D C2 76 50 2A 11 44 22
       AE 50 E5 D1 21 FF FF C3 A6 50
END
```

CUT THOSE HIGH TELEPHONE BILLS

This almost sounds like one of those "Good News-Bad News" jokes. The good news is that you can limit your monthly bill for computer-to-computer calls to $25 a month, no matter how many calls you make. The bad news is that at present, you can only do this if you live in one of twelve major U.S. cities! Hopefully, this service will expand to cover more areas of the country in the not-too-distant future (with my luck, it will probably get into Sault Ste. Marie in about the year 2025...). Don't go away if you don't live in one of the twelve major cities, because I'm going to reveal a little-known way that you can make use of this service to cut your long distance phone bills even if you <u>don't</u> live in one of those major cities. It will only be cost-effective if you do a LOT of telecommunicating, but for some of you, it could be an interim solution to those high bills (particularly if you live fairly close to one of the twelve major cities).

But first, more about the service itself. It is called PC PURSUIT and is being offered by GTE Telenet, and they have established a free-access bulletin board that you can dial up for more information. The BBS number is (800) 835-3001. The following information was downloaded from that BBS, and should give you a little better idea of what is being offered. Thanks to George Matyaszek, SYSOP of the Chicago Greene Machine BBS (his BBS phone number is (312) 622-4442) for tipping me off to this service.

<u>In Pursuit of almost anything:</u>

For just $25 a month, you can establish an unlimited number of connections to free telephone-accessible databases, bulletin boards and other communicating PCs in 12 major metropolitan areas. The PC PURSUIT service can save you up to 75 percent over long-distance telephone charges for your home PC. It's the first cost-effective alternative for home PC users who have been using standard public telephone services to make long-distance connections.

With PC PURSUIT, you can explore a wealth of free resources and even discover more uses for your PC. In any of the locations accessible to PC Pursuit, you have the ability to:

- Call and send real-time information to other communicating PCs.
- Access public bulletin boards to exchange messages. (GTE Telenet is not associated with these bulletin boards or their services.)
- Access free databases for information that can be entertaining and educational.
- Download/upload public domain software.
- Shop and advertise in electronic catalogs.

<u>Service Hours and Billing</u>

The PC PURSUIT service is provided from 6 pm to 7 am Monday through Friday and on weekends from 6 pm Friday to 7 am Monday (local time).

The more you use the service, the more cost efficient it becomes!

A PC PURSUIT subscription includes a one-time registration fee of just $25. Then, you pay a flat monthly fee of $25 for virtually unlimited use of the service. Each connection can last 60 minutes. You are billed monthly through your VISA or MasterCard account. Payment by check is also offered - if you would like to pay by check, call (800) 368-4215 (voice).

<u>Equipment Requirement</u>

To use PC PURSUIT, all you need are —
- a telephone line
- an auto-answer modem - 300 or 1200 bps
- a terminal or a PC with asynchronous communications software

<u>Parameters:</u>

Communication parameters for your hardware should be set-up consistent with the PC or BBS or host computer you wish to dial. We recommend you use the parameters given below, although several other parameter settings also work fine on PC PURSUIT:

Data Bits: 7
Stop Bit: 1
Parity: Space
Duplex: Full
End of Line: Carriage Return Only

<u>How to Use PC PURSUIT</u>

After you are a registered user, you must do the following to use PC PURSUIT:

a) Use your modem to call the local PC PURSUIT access number.

b) Request the city and destination phone number you wish dialed.

c) Hang-up and then receive your call-back.

d) Await the connection and out-dial; then proceed just as if you had dialed the destination phone number yourself.

<u>Example:</u>

You live in New York and have an area code and phone number of (212) 123-4567, and you wish to call a bulletin board in San Francisco in the 415 area code at 765-4321.

a) Dial the local access number of 675-3738 to reach PC PURSUIT.

b) When request session begins, answer the questions as in this example:

```
Welcome to Telenet PC Pursuit

Enter your 7-digit phone number (XXX-XXXX): 123-4567

Which city do you wish to call? SAN FRAN

Enter the phone number you wish to call: 765-4321


You are #( ) in the queue. Do you want to wait,
or restart? (Y/N/R)? Y
```

c) When you key in the N(o) to the last question, your call will be disconnected. If you key in a Y(es), you will see the following:

```
Your request is being processed.

Please hang up and wait for your call-back. Good bye!
```

At this point, you should set your modem and/or PC software to the AUTO ANSWER mode.

d) In about 20 seconds, you will receive a call-back from PC PURSUIT. You will see the following banners:

```
This is your call-back. Please stand by ...

Please wait for 3-step call completion.

1) Network request submitted
2) Network connection complete. Placing request call
3) Connection complete
####
```

e) You now are connected to your requested destination city and phone number. If you've accessed a public bulletin board, most require that you enter two carriage returns or a Ctrl-C.

f) To disconnect from the PC PURSUIT service, just disconnect by hanging up from your communication software, modem and/or PC.

g) Customer Service - If you have any difficulty using PC PURSUIT, call Telenet Customer Service at (800)336-0437.

<u>PC PURSUIT Cities and Access Numbers</u>

CITY	AREA CODE SERVED	LOCAL ACCESS NUMBER	CITY ACCESS CODE
Atlanta	404	584-2873	Atlanta
Boston	617	423-0547	Boston
Chicago	312	565-3927	Chicago
Dallas	214	651-7094	Dallas
Denver	303	671-5146	Denver
Detroit	313	961-9555	Detroit
Houston	713	227-5742	Houston
Los Angeles	213	624-6062	LA
New York	212	675-3738	New York
Philadelphia	215	574-0613	Philly
San Francisco	415	398-1134	San Fran
Washington D.C.	202	659-2863	Wash DC

That ends the downloaded information, and if you have read carefully, you may have concluded that there is no way that you can make use of this service if you live anywhere outside the local calling area of one of the twelve major cities. After all, PC PURSUIT won't dial you back if your number is a toll call, right? That's correct, so what YOU have to do is to arrange to have a local number for PC PURSUIT to dial. Enter a little-known telephone company service called REMOTE CALL FORWARDING (RCF). This service operates in the exact same manner as CALL FORWARDING on your regular telephone (if that service is available in your area), with two exceptions! One, calls are ALWAYS forwarded from the RCF number, and two, there is no actual telephone connected to the RCF number - as I've just said, ALL calls are forwarded to another location.

Now some of you may not even know what regular Call Forwarding is, so let me explain. If your telephone is served by a modern (electronic) telephone exchange, you can request Call Forwarding service (at an extra monthly charge). This gives you the ability to have incoming calls to your telephone automatically transferred to another number. For example, let's say you have a business but want callers to be able to reach you at home during the evening hours. When you leave your place of business at the end of the day, you dial a special code followed by your home telephone number, after which all calls to your business phone are automatically "forwarded" to your home phone. When you return to your office the next morning, you dial a different special code to cancel the call forwarding so that you can once again receive your calls at your place of business.

REMOTE CALL FORWARDING, as stated above, is exactly the same except that calls are ALWAYS forwarded to another number, and the RCF number does not run into a home or business - it exists only in the "mind" of the telephone company computer. The service is intended for business users that wish to provide a "local" telephone number so that people in a specific locality (which is outside the normal "local" calling area of the business) can call that business toll-free. However, in most areas there is probably no regulation which says that a person cannot have a residence RCF line (at the lower residential telephone rate) - the local phone company may never have done it before, but that doesn't necessarily mean that it can't be done!

One further point - calls can be forwarded to a local or long distance number. If a call is forwarded outside the local calling area of the RCF number, the toll charges for that portion of the call are billed to the RCF number, not to the calling party. Now, note that in some cases the calling party (the caller who dials the RCF number) and the party who ultimately receives the call may be far enough apart that the call would be considered a toll call if placed in the normal manner, but if the RCF number is located in a telephone exchange that is somewhere midway between the two points, it may be that no toll charge will be levied! Why? Because, for billing purposes, the call is actually treated as two calls - one call from the calling party to the RCF number (for which the calling party pays, if that portion of the call is a toll call) and another call from the RCF number to the ultimate destination of the call (for which the RCF subscriber pays, if that portion of the call is toll).

Now, all that PC PURSUIT cares about is that it's dialing a local number (within the local calling area of the PC PURSUIT node). Let's take two different cases and see how Remote Call Forwarding can be used to gain access to PC PURSUIT and save money. The two cases that we will discuss are: 1) A user who lives just outside the PC PURSUIT local calling area, 2) A user who lives at some greater distance from the PC PURSUIT node.

1) A user who lives just outside the PC PURSUIT local calling area will often find that there is a telephone exchange that is a local call for both him and the PC PURSUIT node. This will usually be a suburban exchange located between him and the central city area, and there may even be more than one exchange that would qualify. If there is such an exchange and RCF is available in that exchange, the user would simply apply for an RCF number in that exchange and have the calls forwarded to his phone. PC PURSUIT would dial the RCF number (a local call) and the RCF number would forward the calls the user's phone (also a local call). Thus, the additional monthly cost is only that of the RCF line. This setup has an added benefit in that others in the central city area may now make calls to the RCF subscriber on a toll-free basis.

If there is such an exchange (that is local for both the user and PC PURSUIT) but remote call forwarding is not offered there, it may be possible to have two regular phone lines run into a friend's home or office in that exchange area, and to use a device called a "call diverter" to connect the two (in fact, if the exchange is operated by a small, independent phone company, you may be able to work out an arrangement to have the diverter located right on the phone company's premises). This is a very poor solution, however, because most call diverters do not have internal amplification and thus the signal will be degraded, perhaps to the point of making it unusable. Nevertheless, if you know what you're doing, this might be a solution for a stouthearted few.

2) A user who lives at some greater distance from the PC PURSUIT node (too far away to have an overlapping local calling area) may find it advantageous to get an RCF line anyway, and have calls forwarded to his phone. There will be a per-minute toll charge for each call, but it will almost always be much lower than dialing distant computers directly (especially if you call during the discount rate periods). The RCF line should be placed in an exchange that is within the local calling area of the PC PURSUIT node. Be aware that if the RCF line is run out of a suburban exchange rather than a "downtown" exchange, the fixed monthly service rate may well be lower. Why? Because local rates are often based on how many telephones are within the local calling area of an exchange. Suburban exchanges often have a smaller local calling area than the central city, thus the monthly rate is lower. Just don't pick an exchange so far out in the suburbs that you get out of PC PURSUIT's local calling area.

It may be worth your time and effort to determine if any of the suburban exchanges are served by a different telephone company than the central city, since there will almost certainly be a variation in monthly rate (and other charges) if that is the case. Remember, your RCF line can be put in any exchange that is a) a local call for PC PURSUIT, and b) offers the Remote Call Forwarding service. For maximum savings, determine which exchanges meet both of those qualifications, then determine the per minute toll rate from each such exchange to your location. Then, if more than one exchange has the lowest per-minute rate to you, check to see if those exchanges are served by different local phone companies and if so, pick the one with the lowest monthly service rate.

Also, there is a way you may be able to save money on the per-minute toll charges. Depending on how far away you are from the PC PURSUIT major city, and also depending on any applicable state and federal regulations, you may be able to choose which long distance carrier will serve your RCF line (actually, you should be able to choose this anyway, but if you are in the same state (or in some states, the same area code) as the PC PURSUIT node, the local phone company may handle the entire call (and collect the toll charges) anyway. It all depends on how your state Public Service Commission feels about competition among long distance companies). If you have a choice, you want to choose a carrier that offers the best discount to your location (and remember, your calls will only be forwarded to ONE location, so that is the only location you will want rate comparisons for) AND that has quality circuits from the PC PURSUIT node to your location (most MODEMs don't work well on trashy circuits). Be sure to take the "volume discounts" offered by some carriers into consideration when comparing rates. Also, keep in mind that most major metropolitan areas are served by a large number of competing long distance carriers, so it is likely that you will be able to shave your per-minute charges by a good percentage, just by shopping around. One caveat - the long distance service you pick must offer "Dial 1" access in your RCF exchange. Remote Call Forwarding is not designed to access carriers that require you to dial a seven-digit number, wait for a second dial tone, and then dial your account number and the number you want to call.

In both of the above cases, you will first have to dial PC PURSUIT the normal way (i.e., as a toll call) to establish the connection. But then, PC PURSUIT will call you back on your RCF line, the call will be forwarded to you, and you can start saving money. Be sure to take everything into account - the monthly charges for PC PURSUIT and the RCF line, plus your estimated toll charges for calls from the RCF number to you, when deciding if the use of an RCF line will be cost-effective for you.

TAS / NORTHERN BYTES ELECTRONIC MAIL ADDRESSES

MCI Mail ID - Northern Bytes:	102-7413
MCI Mail ID - The Alternate Source:	109-7407
TELEX - Northern Bytes:	6501027413
ANSWERBACK:	6501027413 MCI
TELEX - The Alternate Source:	6501097407
ANSWERBACK:	6501097407 MCI
Compuserve EasyPlex:	72167,161
Delphi Mail:	TASIO

HINTS ON READING TRS-80 COLOR COMPUTER DISKETTES UNDER NEWDOS/80

Downloaded from Apparat's BBS by Bob Seaborn

A few people have requested the ability to read diskettes from the TRS-80 Color Computer (Coco) by NEWDOS/80. This disk is rather foreign to NEWDOS/80, but is close enough to get by.

The Coco disk does not contain anything close to a BOOT/SYS in the way NEWDOS/80 does because the entire DOS resides in the Coco ROM pack / drive adapter. The directory is located on track 17. Radio Shack gives a fairly detailed description of the Coco directory structure as well as disk organization. This discussion will not get involved with use of the directory other than what has just been stated. One could write a program for NEWDOS/80 that would allow for file access of the Coco disk, but we did not see that much call for it, so we didn't implement the idea.

The Coco disk is organized in 35 tracks, 18 sectors per track, double density. It numbers the sectors from 1 to 18 instead of 0 to 17 as does NEWDOS/80. Also, the tracks are numbered from 1 to 35, where NEWDOS/80 numbers these from 0 to 35. Because of these differences, we have to adjust the PDRIVE appropriately in order to access all of the data sectors. On the Model III/4, we use the following PDRIVE:

```
TI=AI,TD=E,TC=35,SPT=18,...
```

The TI flag "I" is used to tell NEWDOS/80 that we have a diskette with sectors numbered from 1 instead of 0. The same PDRIVE is used for a Model I except for the TI specification. Use TI=CI (for Percom, Aerocomp, etc. type double density adapters), TI=DK (for Radio Shack double density adapter; see zap # 76 to implement), or TI=EI (for LNW 5/8 doubler). The TSR specification is not listed as this is dependent on your drive, so just use what you normally use. The GPL, DDSL, DDGA specifications are not listed since we are not accessing the directory. Just use standard numbers here as in examples #1 or #2 on page 2-38 of the NEWDOS/80 manual. If you were to write a special program that would access files via the directory of the Coco diskette, it's doubtful that these would be useful. The Coco directory sectors are not marked as "read protected" as are the NEWDOS/80 directory sectors.

Activate the PDRIVE in drive 1 to allow access to data sectors of the Coco diskette with the following:

```
PDRIVE,0,1,TI=AI,TD=E,TC=35,SPT=18,A
```

With modifications to TI specification for Model I as discussed above.

Use SUPERZAP to examine the diskette. You can use either the "DD" (display disk sector) mode or the "DTS" (display track sector) mode of SUPERZAP. Here, we will access the sectors as if they were numbered from 0 to 17 from each track instead of 1 to 18. Also, we will access the track as if they were numbered from 0 to 34 instead of 1 to 35. The PDRIVE specification adjusts the sector and track numbering for us. Attempting to access track number 35 (actual 36 to the Coco) will result in either a read error (as there is no data to be read) or track (or sector) out of range error. Go to the third sector of the directory to find the first file entries. You may wish consult the Radio Shack documentation to figure out how to find the actual data on the disk from the directory information or just randomly hunt down the file data on the disk. It seems that the Coco allocates file space outwardly from track 17 instead just upwardly from track 0. The Coco file allocation (granule) is in blocks of 9 sectors at a time. To find the start of a file just look at sector 0 or sector 9 of each track.

If you want to bring BASIC programs over to the Model I/III/4, you should have the Coco save the program in ASCII format first. Use: SAVE"filename",A to do this. The same applies for the opposite direction; have the Model I/III/4 save the file in ASCII.

Once you have located the data file on the Coco diskette, determine the exact length and location of the file on the disk. You may want to write down all of the sector numbers related to the file that you want to transfer. Be sure to use relative disk sector number and not track/sector numbers. Count the number of sectors that are of interest. Under NEWDOS/80 Ready (or MINI-NEWDOS/80), use the CREATE command to create a receiving file on the NEWDOS/80 diskette of the same size. This file will receive the data. For example the length of the Coco file is 7 sectors in length.

```
CREATE,SAMPLE/DAT,REC=7
```

Use SUPERZAP "DFS" mode to map out the drive relative sectors (DRS) for all (7, in the above example) sectors allocated to the file (SAMPLE/DAT). Then use CDS (copy disk sectors) to copy the Coco file's sector data to the NEWDOS/80 receiving file's data sectors. Use caution, if you accidentally copy the data to the wrong place, other good data could be destroyed.

If you want to make a copy of a Coco diskette on the Model I/III/4, NEWDOS/80 can do it easily. Set up the same PDRIVE as discussed above. Then, place a blank diskette in drive 1 and enter:

```
COPY 0,1,,FMT,BDU,SPDN=1
```

Follow the disk mounting prompts carefully. When the format and copy sequences are done, your duplicate disk is ready for use.

We were using a diskette from a Color Computer II disk system. We do not know for sure if this applies to the original Color Computer disk system as well or not, but find it hard to believe that Radio Shack would have gone into any major changes to the system.

SPEEDY SERVICE FOR NORTHERN BYTES READERS IN AUSTRALIA AND NEW ZEALAND!

If you live in Australia or New Zealand, you can now place orders with The Alternate Source and get much faster service on most items, including issues of NORTHERN BYTES. If you have a Telex machine, you may contact us directly using our MCI/WUI Telex number, 6501027413 MCI. Otherwise, simply place your order with Tony Domigan, and he will forward the order to us by electronic mail, which will save you up to a week over conventional airmail. Only VISA or MASTERCARD orders can be accepted in this manner.

Tony may be contacted in one of the following ways:

1) Mail - the address is: Tony Domigan, PO BOX 150, Thomastown, Victoria, 3074, AUSTRALIA.
2) Phone: (03) 466-1738
3) TELEDATA (TAB) IDENT : DOMIPOBOWINN
4) VIATEL IDENT :346617380
5) MCI Mail ID: 254-5121
6) U.S.A. Telex address (MCI/WUI network): 6502545121 MCI

The latter two methods are mostly given for the benefit of our North American readers, who may wish to question Tony on one of the articles he has written for NORTHERN BYTES!

Please note that we are at this time experimenting with a new mailing service that will permit us to give airmail service to all of our regular mailings of NORTHERN BYTES, without the $1.00 surcharge we have had in the past. This applies to current issues only, not back issues (which will still be charged at $3.00 each for air delivery).

A DOUBLE SIDED BOOTING DISK UNDER LDOS

by Frank Marten

[Reprinted from the ADELAIDE MICRO-USER News (produced by the ADELAIDE MICRO-USER Group, Inc., 36 Sturt Street, Adelaide, South Australia 5000).]

I recently had an enquiry about how to create a double sided, double density LDOS system disk for the Model I. This resulted in my having a closer look at creating other kinds of DOS disks for the Model I and the Model III.

When backing up LDOS system disks, you may be aware that if you are using 5.1.4, it begins allocating from track zero and works its way up until all files have been copied across (this applies to a backup reconstruct only). On 5.1.3, the allocation is random which will result in the files dispersed all over the disk. Both types of backup reconstructs result in a rather poor system disk. One has its system files at the beginning of the disk mixed with other files, the other method results in any future saving of files to take longer as all the holes need to be filled.

As the steps involved to create an optimum disk require several patches to SYS8/SYS and an organised approach to the files to be copied, the solution to the problem is to create a Job Control Language file which will back-up the disk to whatever format is required.

This is exactly what I have done. The JCL listing provided will create a system disk for the Model I or III, in either single or double sided format and double density. As I have used the method of passing parameters to the JCL on entry, it is easily adaptable to add as many other parameters as you like (as long as they can fit on one line). An example of this might be to allow an option for the density of the destination disk. This would be quite simple but if backing up from a double density to single, you may need to get rid of the global backup of all visible files. Other options could be to allow the passing of the source and destination drives, entering of the master password or whatever other feature you wish.

The JCL file I wrote will allow you to specify the disk name, the number of sides and the model of the machine you are using. The variables used for this are N1 = disk name, S = 1 (will give you a single sided disk) and M3 if it is a Model III. The defaults are N1 = ldossys, sides = 2, model = Model I, and source disk = drive zero, destination drive = one.

One very important point with this JCL is that the format will occur even if there is data on the destination disk. It is for this reason that I made the JCL give audible prompts before it proceeds with the format. It also gives the beeps when user input is required such as after format to ensure the disk was OK and a continuous two tone tune when it has finished to let you know the job has been completed. The reason for this is because it can take more than twenty minutes to complete the process.

If the disk created was a Model III or it was single sided, the JCL will not be required until the next time you wish to create a special disk as QFB will backup the disk much quicker. If you created a double sided, double density Model I system disk, you will not be able to use QFB as it will abort on track zero. Super Utility will also abort on track zero because it will try to copy thirty-six sectors when there are really only ten. You can get around this by skipping the sectors marked as bad by SU+.

When you have completed the backup, on the Model I (double density and double sided disks), you will have to execute LOG :0 and then configure the disk how you like it followed by SOLE2 :0. The LOG command will allow you to switch the new disk into drive zero if your original disk was not double sided. This would also be necessary for a Model III except you do not execute SOLE2.

I have also provided a configuration JCL which will allow you to specify the name of a KSM file. The default is SETKSM/KSM. You would enter this by 'DO CONSET (K=FILESPEC)'.

DOS DISK CREATION JCL

```
%1f.                DOS Disk Creation
.                   by
.                   Frank Marten.
.
. If you did not specify the target disk name & model type
. (m3 for model 3) and sides (s=1) on entry to the JCL,
. Disk name = LDOSSYS, SIDES=2 and model = 1 .
.
.                   i.e. do doscreat (n1=mydisk,s=1,m3)
.
. * NOTE *          The DOS disk must not be write protected!
.
. Programs required on DOS disk are :
.                   ') Sole1 and Sole2 (if model 1)
.                   2) patch/cmd
.                   3) normal sys files
.                   4) normal utilities (backup,format etc)
.                   5) log/cmd (if double sided)
.
.   ** Warning, there is no turning back from FORMAT **
//if -n1
//assign n1=ldossys
//end
//if -s
//assign s=2
//assign t1=11
//assign t2=16
//else
//assign t1=0f
//assign t2=18
//end
.
//alert 1,0
//pause %1dPlace target disk in drive 1 and press <ENTER>.
.
format :1 (name="#n1#",q=n,dden,cyl=40,sides=#s#,abs)
.
//alert 1,0
//pause %1d format O.K.? <ENTER> if yes, <BREAK> if not.
//if -m3
sole1 :1
//end
patch sys8/sys.system (d00,ff=#t1#)
backup sys0/sys:0 :1 (s,q=n)
backup sys6/sys:0 :1 (s,q=n)
patch sys8/sys.system (d00,ff=#t2#)
backup sys7/sys:0 :1 (s,q=n)
patch sys8/sys.system (d00,ff=#t1#)
backup sys:0 :1 (new,q=n,s)
patch sys8/sys.system (d00,ff=01)
patch sys8/sys.system:1 (d00,ff=01)
backup lbasic:0 :1 (q=n,i)
backup /cmd:0 :1 (q=n,i)
backup /dvr:0 :1 (q=n)
backup /flt:0 :1 (q=n)
backup :0 :1 (q=n,i,new)
backup :0 :1 (new)
%1f.
.                   Steps to complete.
.
.                   1) type LOG :0 and switch disks when prompted
.                   (if double sided)
.                   2) configure your new disk
.                   3) type SYSTEM (SYSGEN)
.                   4) execute SOLE2 :0 (if Model I)
.
.                   Press <ENTER> to exit.
//alert (1,0,7,0)
//exit
```

CONFIGURATION JCL

```
%1f                 Set up Configuration
.
. If you did not enter the ksm file on entry, the default
. is setksm/ksm
.
.                   i.e. do conset (k=filespec)
.
//alert 1,0
//flash %1d Press <ENTER> when disk is ready.
.
system (drive=0,cyl=40,step=0,delay=on)
system (drive=1,cyl=40,step=0,delay=on)
system (drive=2,cyl=40,step=0,delay=on)
system (bstep=0)
system (blink,small)
set *ki ki (j,t,d=25,r=1)
//if -k
//assign k=setksm/ksm
//end
filter *ki ksm #k#
.
//alert 1,0
//FLASH %1d Press <ENTER> to exit and then sysgen the disk.
//exit
```

If using LDOS 5.1.3, apply the following patch to SYS8/SYS before using the DOS creation JCL:

```
PATCH SYS8/SYS.SYSTEM (D00,FE=2E 01 00 00 00 00)
```

PATCHES FOR MODEL 4 USERS

Patches and zaps seem to be ever popular with our readers, so here's a few more. These came from various sources, but probably originated with Radio Shack and/or Logical Systems, Inc. As always, these patches are untested by us here at NORTHERN BYTES, so apply them at your own risk, and please, to a backup of your master disk only!

The first series of patches are intended to correct a possible problem with disk I/O in both LDOS 5.1 and TRSDOS 6, when running on later versions of the Model 4/4P. This problem is caused by Radio Shack's change to the 1770 and 1773 Western Digital disk controller chips. These chips are not 100% software compatible with the controller chips used in previous models, and require larger minimum gaps in data and address fields.

These chips are present in the newer Model 4 and in all Model 4P computers, specifically on the Gate Array Logic boards. These patches should work with all of the disk controller chips used in the Models 4 and 4P (old or new), so you can apply them to your DOS and for use with any Model 4 series computer. The patches will also be made a part of the next releases of TRSDOS 6 and LDOS.

Please note that the problem is with the disk format itself as used on this type of disk controller, so you should format a batch of blank diskettes with the new version of FORMAT, and then BACKUP all your existing diskettes to the newly formatted ones.

PATCH FOR TRSDOS 06.02.00

```
. FORMATC/FIX - 05/06/85 - by —jjkd—
.
. This patch is for TRSDOS 06.02.00 Level AN *ONLY*
.
. This patch to FORMAT will increase the gaps to the nominal requirements
. of the 1770 and 1773 chips. This is necessary as the required gaps are
. larger than for the previously used 1771, 1791, 1793 series chips.
.
. Use BUILD to create this patch file, and then apply using the command:
. PATCH FORMAT/CMD.UTILITY FORMATC/FIX
.
D05,7A=20
F05,7A=14
D05,99=4E 17
F05,99=FF 11
.
. End of patch
```

PATCH FOR LSQFB/CMD

```
. LSQFBC/FIX - 05/06/85 - by —jjkd—
.
. This patch is for LSQFB/CMD from LSI's LS-QFB/COMP package. It is already
. present on Radio Shack's TRSDOS 6.2 Utilities package.
.
. This patch to QFB will increase the gaps to the nominal requirements
. of the 1770 and 1773 chips. This is necessary as the required gaps are
. larger than for the previously used 1771, 1791, 1793 series chips.
.
. Use BUILD to create this patch file, and then apply using the command:
. PATCH LSQFB/CMD.UTILITY LSQFBC/FIX
.
D08,CD=20
F08,CD=14
D08,EC=4E 17
F08,EC=FF 11
.
. End of patch
```

PATCH FOR LDOS 5.1.4

```
. FORMATC/FIX - 05/06/85 - by —jjkd—
.
. This patch is for LDOS 5.1.4 Model 3 and Model 4 in Model 3 mode *ONLY*
.
. This patch to FORMAT will increase the gaps to the nominal requirements
. of the 1770 and 1773 chips. This is necessary as the required gaps are
. larger than for the previously used 1771, 1791, 1793 series chips.
.
. Use BUILD to create this patch file, and then apply using the command:
. PATCH FORMAT/CMD.RRW3 FORMATC/FIX
.
D05,8D=20
D05,AC=4E 17
.
. End of patch
```

PATCH FOR QFB/CMD FROM LDOS 5.1.4

```
. QFBC/FIX - 05/06/85 - by —jjkd—
.
. This patch is for QFB/CMD from LDOS 5.1.4 *ONLY*
.
. This patch to QFB will increase the gaps to the nominal requirements
. of the 1770 and 1773 chips. This is necessary as the required gaps are
. larger than for the previously used 1771, 1791, 1793 series chips.
.
. Use BUILD to create this patch file, and then apply using the command:
. PATCH QFB/CMD.RRW3 QFBC/FIX
.
D0A,CE=20
D0A,ED=4E 17
.
. End of patch
```

The next two patches are for Model 4/4P users who want to make the fullest use of the sound board in their machines.

PATCH FOR CLICK/FLT

```
.This jcl patches CLICK/FLT for a better sound.
.Patch can be YANKED
.It does NOT check to see what is at the listed memory location
.but simply patches it.
PATCH CLICK/FLT.FILTER (X'247B'=28 27)
RESET *KI
SET *CK CLICK/FLT
FILTER *KI *CK
//EXIT
```

PATCH FOR LDOS 5.1.4 (JCL ALERT)

```
.This patch file for LDOS 5.1.4 changes the JCL
.ALERT Macro output from the cassette port to
.the sound board in the model 4
.
.to use type => patch sys11/sys.system alert/fix
.
D02,7C=93
D02,82=93
.End of patch
```

Finally, here are two sets of patches for the Model 4 versions of Radio Shack programs:

PATCH FOR ALDS

```
.Patches to the TRSDOS 6 ASSEMBLY LANGUAGE DEVELOPEMENT SYSTEM
.These patches are from Radio Shack Customer Service Bulletins nos.
.2012-01, 2012-02, 2012-03, 2012-04
.This will bring the current version of the ALDS to 03.02.03.
.ALEDIT is the only module that will contain the new version number.
.
.Fixes the Wait on Error switch for the assembler.
.Fixes the Hold Key Function.
PATCH ALASM (D06,53=CD 4A 3D 00:F06,53=38 02 18 03)
PATCH ALASM (D08,01=5C:F08,01=5F)
PATCH ALASM (D09,39=08 EF C0:F09,39=65 EF FD)
PATCH ALASM (D09,40=FE 80 28 CE FE 60 C0 FD:F09,40=CB 0A 46 20 CD FD CB 0A)
PATCH ALASM (D09,48=CB 0A 8E 3E 01 EF FE 80:F09,48=4E CB FD CB 0A 8E 3E 65)
PATCH ALASM (D09,50=28 C0 C9 00 00 00 00 00:F09,50=EF FD CB 0A 46 20 0B FD)
PATCH ALASM (D09,58=00 00 00 00 00 00 00 00:F09,58=CB 0A 56 28 F1 FD CB 0A)
PATCH ALASM (D09,60=00 00:F09,60=96 C9)
PATCH ALEDIT/CMD (D00,31=31:F00,31=30)
.
.Defeats the blank page feed after a listing. (Optional)
//KEYIN Do you want to apply this patch  (1-Yes 2-No)
//1
PATCH ALASM/CMD (D09,80=00 00 00 00 00:F09,80=06 0C CC 41 3A)
///
.
.Fixes cursor in ALBUG M option.
PATCH ALBUG/CMD (D0E,12=C3 5C:F0E,12=E5 C5)
PATCH ALBUG/CMD (D0E,18=E4:F0E,18=4F)
PATCH ALEDIT/CMD (D00,31=32:F00,31=31)
.
.Keeps the characters in the first column from disappearing
.when the cursor is moved.
```

```
PATCH ALEDIT/CMD (D05,46=C3 E1 3E:F05,46=C5 D5 E5)
PATCH ALEDIT/CMD (D00,31=33:F00,31=32)
.
//EXIT
```

PATCH FOR SUPERSCRIPSIT 1.01.00

```
. PATCHES TO UPDATE SUPERSCRIPSIT 1.01.00 ON TRSDOS 6.2
.   TO LATEST VERSION, WHICH IS 1.01.03
.
. UPDATE 1.01.00 TO 1.01.01
. (PROBLEM: A block deletion that spans more than one extent
.  on the disk will cause the system to lock up.)
.
PATCH SCR32/CTL (D03,8F=AD 40:F03,8F=98 6F)
PATCH SCR17/CTL (D01,C1=31:F01,C1=30)
.
. UPDATE 1.01.01 TO 1.01.02
. (PROBLEM: A Document returns to the screen in reverse video
.  after printing if it contains headers and footers.)
.
PATCH SCRIPSIT/CTL (D0A,B5=21 00 03 22 3B B6:F0A,B5=00 00 00 00 00 00)
PATCH SCRIPSIT/CTL (D0A,CE=C3 F2 6D:F0A,CE=11 D4 C1)
PATCH SCRIPSIT/CTL (D0B,61=21 00 03 22 3B B6:F0B,61=00 00 00 00 00 00)
PATCH SCRIPSIT/CTL (D3E,D8=32 FF 6D CD 57 8D AF:F3E,D8=FE F1 C2 9C 99 CD 0F)
PATCH SCRIPSIT/CTL (D3E,DF=32 FF 6D C9 3A FF 6D:F3E,DF=44 C3 AF 99 C0 CD 0F)
PATCH SCRIPSIT/CTL (D3E,E6=87 C2 C8 3A 11 D4 C1:F3E,E6=44 3E 01 C3 B2 9B 00)
PATCH SCRIPSIT/CTL (D3E,ED=C3 A5 3A:F3E,ED=00 00 00)
PATCH SCRIPSIT/CTL (D3F,6D=E7 6D:F3F,6D=57 8D)
PATCH SCRIPSIT/CTL (D3F,91=12:F3F,91=10)
PATCH SCR17/CTL (D01,C1=32:F01,C1=31)
.
. UPDATE 1.01.02 TO 1.01.03
. (PROBLEM: A "close fault" error message will occur when attempting
.  to exit the program if HELP/CTL has been deleted from the disk.)
.
PATCH SCRIPSIT/CTL (D3E,43=21 25:F3E,43=00 00)
PATCH SCRIPSIT/CTL (D3E,49=83 7E FE FF 28 0C 18:F3E,49=00 00 00 00 00 00 00)
PATCH SCRIPSIT/CTL (D3E,50=04:F3E,50=00)
PATCH SCRIPSIT/CTL (D3E,52=18 F3 00:F3E,52=21 25 83)
PATCH SCRIPSIT/CMD (D01,0C=33:F01,0C=30)
PATCH SCR17/CTL (D01,C1=33:F01,C1=32)
```

DATECOMP/ASM

A date compression routine by Darren R. Besler

[This assembly language source code listing is reprinted from the Winnipeg (Manitoba, Canada) Micro-80 User Group Newsletter, with slight changes in formatting to make it fit within the Northern Bytes column width. If you have ever wanted to sort by date, or to store a date in only two bytes of memory, this routine may offer a partial solution to your problem!]

```
          00100 ;DATECOMP/ASM
          00110 ;A date compression program for Z-80 based microcomputers.
          00120 ;
          00130 ;Written By: Darren R. Besler
          00140 ;Date:       March 21, 1985
          00150 ;
          00160 ;This program/subroutine compresses a 6 byte ASCII date
          00170 ;field into a 2 byte integer.  This program is a spin off
          00180 ;of the BASIC program written by John Allison which was
          00190 ;published in the March 9, 1985 issue of 'THE WINNIPEG
          00200 ;MICRO-80 USERS GROUP NEWSLETTER'.
          00210 ;
          00220 ;The ptr to the date is passed in the HL register, and the
          00230 ;compressed date is returned in the HL register. All other
          00240 ;registers are preserved by this subroutine.  Very little
          00250 ;error checking is done and NO error check should be
          00260 ;assumed.
          00270 ;
          00280 ;The date format accepted is as follows: 'MMDDYY', where
          00290 ;M,D, and Y are the ASCII representation of the digits in
          00300 ;the date.
          00310 ;
          00320 ;The date is encoded as follows:
          00330 ;                  YY     MM    DD
          00340 ;                xxxxxxx xxxx xxxxx
          00350 ;               0 <- 16 bits ->    0
          00360 ;An eg.    March 19, 1985 ('031985') will result in the
          00370 ;following word:
          00380 ;             1010101001110011 = 0AA73H = 43537D
          00390 ;This results in an unsigned integer. This could allow for
          00400 ;a very quick sort on a date field along with the space
          00410 ;reduction.
          00420 ;
5200      00430        ORG     5200H
          00440 ;
5200 F5   00450 DTCOMP PUSH    AF      ;Save registers on the stack
5201 C5   00460        PUSH    BC
5202 DDE5 00470        PUSH    IX
5204 E5   00480        PUSH    HL      ;Place ptr to date string into IX
5205 DDE1 00490        POP     IX
          00500 ;
          00510 ;Get year field.
5207 DD7E04 00520 YY   LD      A,(IX+04H) ;Get 1st character in the year
520A D630 00530        SUB     30H     ;Make it an integer
520C CD4C52 00540      CALL    MUL10   ;Multiply by ten
520F 4F   00550        LD      C,A     ;Save in 'C' register
5210 DD7E05 00560      LD      A,(IX+05H) ;Get second character of date
5213 D630 00570        SUB     30H     ;and make into an integer
5215 81   00580        ADD     A,C     ;add together
5216 CB27 00590        SLA     A       ;Shift into position
5218 67   00600        LD      H,A     ;Place year into 'H' register
          00610 ;
          00620 ;Get Month field.
5219 DD7E00 00630 MM   LD      A,(IX+00H) ;Get 1st character of Month
521C D630 00640        SUB     30H     ;Make it an integer
521E CD4C52 00650      CALL    MUL10   ;Multiply by ten
5221 4F   00660        LD      C,A     ;Save in 'C' register
5222 DD7E01 00670      LD      A,(IX+01H) ;Get 2nd char of month field
5225 D630 00680        SUB     30H     ;Change into an integer
5227 81   00690        ADD     A,C     ;and add together
5228 CB5F 00700        BIT     3,A     ;Check if Month hi order bit set
522A 2802 00710        JR      Z,HI0   ;If not - ignore, else
522C CBC4 00720        SET     0,H     ;Set Hi order bit of Month in 'H'
522E 0F   00730 HI0    RRCA            ;Place 3 least significant bits
522F 0F   00740        RRCA            ;  into 3 hi order bits
5230 0F   00750        RRCA
5231 E6E0 00760        AND     0E0H    ;Mask off 5 least sig bits
5233 6F   00770        LD      L,A     ;And place Month into 'L' reg
          00780 ;
          00790 ;Get Day field.
5234 DD7E02 00800 DD   LD      A,(IX+02H) ;Get 1st char of day field
5237 D630 00810        SUB     30H     ;and make it into an integer
5239 CD4C52 00820      CALL    MUL10   ;Multiply by 10
523C 4F   00830        LD      C,A     ;Save in 'C' register
523D DD7E03 00840      LD      A,(IX+03H) ;Get 2nd char of day field
5240 D630 00850        SUB     30H     ;Make into an integer
5242 81   00860        ADD     A,C     ;and add together
5243 E61F 00870        AND     1FH     ;Mask off 3 most sig bits
5245 B5   00880        OR      L       ;Combine Month and Date
5246 6F   00890        LD      L,A     ;Place back into 'L' register
          00900 ;
          00910 ;The compressed date is now an integer in 'HL' registers
5247 DDE1 00920        POP     IX      ;Pop saved registers off the
5249 C1   00930        POP     BC      ;  stack and return
524A F1   00940        POP     AF
524B C9   00950        RET
          00960 ;
          00970 ;This is a subroutine to multiply a number (0-15) by 10.
          00980 ;The number to be multiplied (the multiplicand) is passed
          00990 ;in the 'A' register and the result is returned in the
          01000 ;'A' register.
524C C5   01010 MUL10  PUSH    BC      ;Save 'BC' register
524D E60F 01020        AND     0FH     ;Make sure multiplicand is 0-15
524F CB27 01030        SLA     A       ;Multiply by 2 by shifting left
5251 4F   01040        LD      C,A     ;Save 2*multiplicand
5252 CB27 01050        SLA     A       ;Multiply by 2 again
5254 CB27 01060        SLA     A       ;Multiply by 2 again
5256 81   01070        ADD     A,C     ;Add 2*multiplicand to product
          01080 ;The final product is in the 'A' register now.
5257 B7   01090        OR      A       ;Set flags
5258 C1   01100        POP     BC
5259 C9   01110        RET
          01120 ;
5200      01130        END     DTCOMP
00000 TOTAL ERRORS

DD     5234   DTCOMP 5200   HI0    522E   MM     5219   MUL10  524C
YY     5207
```

MAKE TRSDOS 1.3 YOUR OYSTEI

A BASIC program by Patrick Sweetman & Ola · Skarsholt

[Patrick and Olaf are members of the Christchurch-80 Users Group (P.O. Box 4118, Christchurch, New Zealand) and this program originally appeared in that group's newsletter. As is sometimes the case with programs published in NORTHERN BYTES, the documentation for this one is in the program's REM statements. By the way, to call the phone number in line 50, U.S. readers would dial 011+64+9+817-8698. Remember the time difference, New Zealand is 18 hours ahead of Eastern Standard Time and 21 hours ahead of Pacific Standard Time, and don't forget to call (800) 874-4000 first to find out when the lowest rates are in effect!]

```
10 REM        OYSTER/BAS
20 REM
30 REM   BY PATRICK SWEETMAN & OLAF SKARSHOLT,
40 REM   203A GODLEY ROAD, TITIRANGI, AUCKLAND 7,
50 REM   NEW ZEALAND.    PH.(09) 817 8698.
60 REM
70 REM      2 OCTOBER 1981
80 REM
90 REM
100 CLS:CLEAR 600:PRINT:PRINT
110 PRINT "THIS PROGRAMME TAKES A NORMAL TRS-80 III,"
120 PRINT "TRSDOS 1.1, 1.2, 1.3, SYSTEM DISK (ALL VERSIONS) "
130 PRINT "AND CREATES ON IT DIRECTORY ENTRIES"
140 PRINT "FOR 'BOOT', 'DIRECTORY', AND 'SYSTEMS'"
150 PRINT "FILES ALREADY PRESENT ON THE DISK."
160 PRINT: PRINT "THEN, FOR EXAMPLE, YOU CAN EDIT"
170 PRINT "THE DIRECTORY WITH 'DEBUG', 'F' FUNCTION,"
180 PRINT "AND THEN, THE SYSTEM IS YOUR OYSTER."
190 PRINT:INPUT "ENTER DISK DRIVE NUMBER ";DD$
200 CLS:PRINT "O.K.,  PLEASE BE PATIENT."
210 DIM BU$(1),FC$,X(9),P$,LL$(18),LM$(18)
220 BU$(0)=STRING$(240,32): BU$(1)=STRING$(32,32)
230 FOR N=1 TO 9: X(N)=256: NEXT N
240 FC$=STRING$(64,32)
250 FOR N=1 TO 18: READ LL$(N): FOR X=1 TO 4: READ A: LM$(N)=
LM$(N)+CHR$(A): NEXT X: NEXT N
260 FOR N=1 TO 63: READ A$: A=ASC(A$)-48: IF A>9 THEN A=A-7
270 A=A*16: A0=ASC(MID$(A$,2,1))-48: IF A0>9 THEN A0=A0-7
280 P$=P$ + CHR$(A+A0): NEXT N
290 CLS: PRINT "CREATING THE DIRECTORY ENTRIES NOW."
300 GOTO 510
310 A=VARPTR(FC$): A=PEEK(A+1)+PEEK(A+2)*256
320 X(4)=INT(A/256): X(5)=A-256*X(4)
330 FOR A=1 TO 7: MID$(P$,A+3,1)=CHR$(X(A)): NEXT A
340 P=VARPTR(P$): DEFUSR0=PEEK(P+1)+256*PEEK(P+2)-65536
350 A=USR(0): IF MID$(P$,3,1)=CHR$(0) THEN ER=0: RETURN
360 ER=ASC(MID$(P$,3,1)): PRINT "***** TRSDOS ERROR CODE ";ER
;" *****"
370 PRINT:PRINT "PRESS   A TO ABORT":PRINT "       C TO CONTI
NUE":A$=" "
380 A$=INKEY$: IF A$="C" THEN 390 ELSE IF A$<>"A" THEN 380 EL
SE STOP
390 CLS: RETURN
400 DATA SYS00,15,00,00,05,DIRECT,18,00,17,06,SYS01,05,00,16,02
410 DATA SYS02,04,00,16,66,SYS03,03,00,16,129,SYS04,03,00,16,161
420 DATA SYS05,08,00,00,163,SYS06,17,00,18,06,SYS07,16,00,01,70
430 DATA SYS08,18,00,19,06, SYS09,15,00,20,05,SYS10,04,00,15,02
440 DATA SYS11,10,00,20,164,SYS12,04,00,15,66,SYS13,04,00,15,130
450 DATA SYS14,02,00,21,97, BASIC,00,00,00,00,CONVERT,00,00,00,0
0
460 DATA 18,0A,00,00,00,00,00,00,00,00,00,00,CD,0B,00,E5
470 DATA DD,E1,DD,36,F3,00,DD,66,F5,DD,6E,F6,DD,74,23,DD
480 DATA 75,24,DD,46,F4,DD,56,F7,DD,5E,F8,DD,66,F9,DD,6E
490 DATA FA,CD,00,00,C8,CD,0B,00,E5,DD,E1,DD,77,CA,00,C9
500 A=VARPTR(BU$(0)): MID$(FC$,4,1)=CHR$(PEEK(A+1)): MID$(FC$,
5,1)=CHR$(PEEK(A+2)): RETURN
510 FOR X=1 TO 16: FC$=LL$(X)+"/SYS:"+DD$+STRING$(40,32): PRIN
T @400,FC$;"      "
520 X(1)=0: X(2)=32: X(3)=68
530 A=VARPTR(BU$(0)): A=PEEK(A+1)+PEEK(A+2)*256
540 X(6)=INT(A/256): X(7)=A-256*X(6)
550 GOSUB 310: X(2)=40:GOSUB 310  :NEXT X
560 X(2)=36: GOSUB 310
570 MID$(FC$,13,2)="XX": MID$(FC$,17,1)=CHR$(17): MID$(FC$,18,1)
=CHR$(31)
580 X(2)=54: GOSUB 310: GOSUB 310: C=80
590 IF MID$(BU$(0),C,1)=CHR$(0) THEN C=C-1: GOTO 590
600 FOR CB = 1 TO 1+INT(C/5): GOSUB 500: GOSUB 310
610 ZX=0: FOR D= 0 TO 4: FOR E= 1 TO 18
620 IF MID$(BU$(0),48*D+6,LEN(LL$(E)))<>LL$(E) THEN 670
630 IF E=17 OR E=18 THEN MID$(BU$(0),48*D+17,2) = MID$(BU$(0),4
8*D+19,2): GOTO 650
640 MID$(BU$(0),48*D+21,4) = LM$(E)
650 IF LL$(E)= "DIRECT" THEN MID$(BU$(0),48*D+1,1)="Z" ELSE MID
$(BU$(0),48*D+1,1)="X"
660 ZX=1: PRINT @400,LL$(E);"          ": MID$(LL$(E),1,1)="*"
670 NEXT E: NEXT D: IF ZX=0 THEN 690
680 GOSUB 500: X(2)=69: GOSUB 310: X(2)=57: GOSUB 310: X(2)=54
690 NEXT CB
700 CLS: PRINT: PRINT "O.K.   ALL FINISHED": PRINT
710 PRINT "THE FILES ARE STORED AS 'INVISIBLE  SYSTEM  FILE
S'"
720 PRINT "WITHOUT PASSWORD PROTECTION. "
730 PRINT: PRINT "THE 'DIRECT/SYS' SHOULD NOT, UNDER ANY"
740 PRINT "CIRCUMSTANCES BE KILLED, SO THIS FILE"
750 PRINT "HAS BEEN GIVEN ACCESS LEVEL 2 (RENAME, UPDATE,
 ETC.)"
760 PRINT: PRINT
770 END ' THAT IS ALL
```

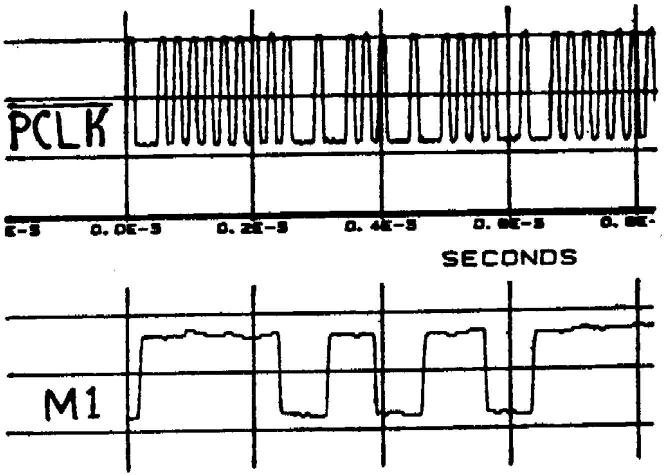
FAST FIX FOR MODEL 4 SLOWS

by Dave Owen, CCS Technical Specialist

[This article is reprinted from Micro Info Exchange, the newsletter of the Cabrillo Computer Society.]

The Radio Shack Model 4 Computer is advertised to operate at a two- or four- megahertz clock rate. Our investigation has revealed the truth of this claim. The Tandy specifications do not, however, state the fact that the master clock does not always "tick". Because of this persistent miss, the machine operates at approximately 3.37 MHz - or nearly one-sixth slower than claimed.

The pre-programmed PAL (programmable array logic in the back panel's U3 socket) logically ties the op-code fetch signal (M1) to the crystal oscillator output (*PCLK). The result is that clock cycles are suppressed during each op-code fetch on those PALs labeled with Radio Shack part number 8075166. A later version PAL has reportedly eliminated this design feature. But most of those sold in 1983 and earlier do run below specs.



This shows component M1 causing PCLK (Primary Computer Clock) in U3 to also output erratically, so that the missing pulses mean, on balance, slower speed.

The clock suppressions aren't needed with the Z80A (the computer was originally designed to operate with a ZILOG Z800 that was never released). If you have this particular PAL, the solution is simple: Remove IC chip U3 by prying it up at each end with a small screwdriver. With a pair of small pliers, bend pin 7 horizontal so it will no longer sit in the socket. Replug this chip back into your board. Presto, a 4 MHz operating speed.

Does this really make a difference? Software, particularly the operating system, is "tuned" to certain operating speed environments. When formatting my 8-inch drives under TRSDOS 6.2, for example, the time required dropped from 8 minutes to just over 2 minutes. Other operations also work more efficiently, such as mathematical calculations. We timed FOR-NEXT loops at 16% faster. Try it, you'll like it. And it's free!

SELF-BOOTING NEWDOS/80 system disk for the MODEL 4P.....FINAL.

by Art Rasmussen

Here is a further modification to the self-booting NEWDOS/80 system disk article that appeared on page 10 of NORTHERN BYTES Volume 6, Number 5. The problem with the other version is that in order to create a copy of the new system disk you had to do a full diskette (type 5) copy. You couldn't do a CBF (type 6) copy. The CBF (Copy By Files) routine copies any user files to the first empty sectors it finds on the destination disk and thus the MODELA/III file would end up being moved from its proper location on the new disk. In order to get NEWDOS to copy the MODELA/III to its proper location I renamed the MODELA/III in the actual NEWDOS directory and called it SYS22/SYS. I then put the FPDE in the fourth slot of the first directory sector (lump 17 relative sector 2). This way NEWDOS's CBF routine will treat it as a system file and during the copy, it will copy it to the same location as it was on the original disk.

PDRIVE setting both drives 0 and 1:

TI=A,TD=E,TC=40,SPT=18,TSR=0,GPL=2,DDSL=17,DDGA=2

STEP 1.

Make a backup of your NEWDOS disk by typing:

```
COPY,0,1,,CBF,/SYS,FMT<CR>
```

From here on I used the utilities in SUPER UTILITY. Make sure you use the N3DR specifier for the configuration table.

STEP 2.

If you did not create a system disk as outlined in step one you will need to make sure that lumps (relative tracks) 30 through 36 are not assigned to any file. The ALLOCATION MAP routine of SUPER UTILITY will tell you this.

The reason for this is that beginning on lump 30, relative sector 6 we will create a dummy directory for the MODEL 4P loader. This sector corresponds exactly to real track 17, sector 0 which is where relative byte 2 on the boot sector says the directory is supposed to be. Notice we will NOT move the real directory from lump 17. We will also copy the MODELA/III file to the sectors immediately following this dummy directory.

STEP 3.

Using SUPER UTILITY, zero out lump 30 relative sectors 6,7,8. Next go to lump 30 relative sector 8 (real track 17 relative sector 3) and beginning at relative byte 0 enter the following:

```
1000 7C00 004D 4F44 454C 4120 2049 4949
9642 9642 3900 1129 FFFF FFFF FFFF FFFF
```

This is the fake FPDE for the MODEL 4P loader. Make sure that this modification begins exactly on the lump, sector, and byte specified. Note we do not need to read-protect these sectors as if they were part of a real directory, nor do we need a HIT sector or a GAT sector, the 4P loader doesn't seem to care.

STEP 4.

Next I used the FILE LOCATIONS routine of SUPER UTILITY to find out where the 57 sectors of the MODELA/III ROM image were located on the MODEL 4 disk. (On mine it began on track 36, sector 12. If you are copying from the MODELA/III file disk, it begins on track 0 sector 4.)

STEP 5.

Using the COPY SECTORS routine of SUPER UTILITY copy the 57 sectors from the MODEL 4 disk to the NEWDOS disk BEGINNING at lump 31 relative sector 2 (this corresponds to real track 17 sector 6). You cannot use the copy file routine of SUPER UTILITY or of NEWDOS to copy the file. They will copy it to the first available sectors on the destination disk and not to the proper location.

STEP 6.

Now go to lump 17 relative sector 2. This should be the first sector of the real NEWDOS file directory entries (FDE). You should see the BOOT/SYS, SYS6/SYS, and SYS14/SYS files listed. Now we will construct an FPDE for the MODELA/III file. However, we are going to call it SYS22/SYS. Begin modification at relative byte 60 and change the two rows of zeros to the following (it is very important that the FPDE begins at relative byte 60. If the file is in another slot, the CBF routine may not treat it as a system file and thus copy it to a different location in which case the new disk will not boot on the 4P):

```
5F20 0000 0053 5953 3232 2020 2053 5953
9642 9642 4000 1E2C FFFF FFFF FFFF FFFF
```

STEP 7.

Use the REPAIR GAT SECTOR routine of SUPER UTILITY to correct the GAT table. This adjusts the GAT table for the new MODELA/III ROM image file and the transferred directory. NEWDOS will treat this as one large file.

STEP 8.

Use the REPAIR HIT SECTOR routine of SUPER UTILITY to correct the HIT table. This places the HIT code for the new SYS22/SYS file in the HIT table.

You may now reboot the disk by pressing the reset key. You do not need to hold any keys down, even if you have just turned the 4P on, the ROM image will load and NEWDOS will boot.

Art Rasmussen
612 West Hillcrest
Keene, Texas 76059
Phone (817) 641-8922

LDOS, TRSDOS 6 AND 1987 - THE WHOLE STORY

[This originated with the LSI Customer Service Department, and since 1987 is only a couple of years away, I feel that many of you might want to know about this now. Please note that certain other DOSes use this same method of date storage, and thus will have similar problems with date stamping of files starting in 1988!]

LDOS (and TRSDOS 6) uses a three bit field to store the year information for the system and in the disk directory for a particular file. The year field therefore has an eight year range, with a base offset (currently) of 1980. Thus, dates ranging from January 1, 1980 to December 31, 1987 can be represented.

LSI is in the process of developing a series of patches to allow a variable base offset other than 1980. These patches will be made available to all registered owners when/if they are finished. This is anticipated before the end of 1985. If the patches become too extensive, or actually require the re-assembly of any part of the operating system, they may instead be made part of a new release of the LDOS system. This would be 5.1.5, and would be available to all registered owners for the normal $10 update fee.

As far as TRSDOS 6 goes, availability of patches and/or a new release of the operating system is up to Radio Shack. If the change is made as a series of patches, it is likely that the patches will be available here.

Of course, because the range of dates that can be represented can't exceed eight years (remember, only three bits), files marked with dates under the "old" date system may show "funny" date information (the date will be advanced from its "old" value by whatever the change in the offset value is).

Sincerely, LSI Customer Service Department

WARNING TO 2400 BAUD MODEM BUYERS

According to an article in the TATSIG (Toledo Area Tandy Special Interest Group) newsletter, the Hayes Smartmodem 2400 is not "Hayes compatible", that is, the internal commands recognized by this MODEM are different from the previous Hayes standard. In fact, at present the Smartmodem 2400 will only operate with Hayes' own Smartcom II program. If you need "Hayes compatibility" you are advised to make sure that the 2400 baud modem you are considering recognizes the Hayes (300/1200 baud) command set. The Multimodem 224 (Multi-Tech Systems, Inc., 82 Second Avenue S.E., New Brighton, Minnesota 55112) is one that is known to recognize the command set of the earlier model Hayes modems, and presumably other such 2400 modems are (or will be) available.

It's nice to know that Tandy isn't the only company that goes around shooting themselves in the foot. I'll bet this "non-Hayes-compatible" Hayes modem turns out to be a major blunder for Hayes, and will probably cost them a good many sales. Not only that, but in the end they'll probably wind up having to retrofit their stock of Smartmodem 2400's to recognize the original Hayes command set. Perhaps the folks at IBM (specifically the PC Jr. people) will be able to offer a little advice, or at least sympathy, when that happens!

MICROSOFT BASIC AND FLOATING-POINT NUMBERS

by Jack Decker

[This article originally appeared in NORTHERN BYTES Volume 2, Number 8 (which is no longer available, so please don't write and ask for a copy).]

I firmly believe that there are thousands of microcomputers that are cranking out reports and data that is incorrect. Only slightly incorrect, mind you, not so much that you'd ever notice in many cases, but still incorrect. If you've ever wondered why things just don't seem to add up properly, you may have been bitten by the floating-point BUG in Microsoft (and many other) versions of BASIC. To see the BUG in action, try typing in this one-line program, then RUN it:

```
10 A=A+.01: PRINT A, A*100, INT(A*100): GOTO 10
```

(Note that if you are not using Microsoft BASIC, the colon separator may not be allowed. Just make a three-line program with one statement per line in that case.) If you are using the ROM BASIC of a TRS-80 Model I or III (or the Model III mode of a Model 4), your display will start out something like this (use the SHIFT-@ key to "freeze" the display):

```
.01              1             1
.02              2             2
.03              3             3
 .               .             .
 .               .             .
.09              9             9
.1               10            9
.11              11            10
.12              12            11
.13              13            13
 .               .             .
 .               .             .
.33              33            33
.34              34            33
.35              35            34
 .               .             .
 .               .             .
.83              83            82
.839999          83.9999       83
.85              84.9999       84
.859999          85.9999       85
.87              86.9999       86
.879999          87.9999       87
.889999          88.9999       88
 .               .             .
 .               .             .
.999999          99.9999       99
1.01             101           100
1.02             102           101
 .               .             .
 .               .             .
4.12             412           411
4.13             413           412
4.14             414           414
4.15             415           415
```

The display starts out as you'd expect, everything going along fine until you reach a certain point (A=.1 in this case). Here we prove that INT(10)=9?! This error with the INT function lasts through a few iterations, then disappears for a while. Next, we prove that the computer can't even add properly (since when does .83 + .01 = .839999?). We then go through a couple of strange iterations where the computer can't decide how to add .01, or how to multiply the result by 100. The INT function seems to settle into a pattern. As our value of A tops 1, the INT function seems to always deliver a value that is low by 1. This continues until A=4.14. Then, suddenly, all values are correct! Keep running the program, however, and you'll find that this happy state doesn't last. Seems the computer can't even add properly, much less multiply.

Now, you might be thinking that using double precision numbers will solve the problem. If you're using a TRS-80, add line 5:

```
5 DEFDBL A
```

to force A to be a double precision variable. Also, change ".01" in line 10 to ".01D" to force the constant to be evaluated as double precision. The problem won't show up as soon, but eventually you'll see:

```
.66                      66                        66
.6699999999999999        67                        67
```

Wait, why does the INT function and the multiplication by 100 still work? Because 100 is still a single precision constant, thus forcing the result to single precision. Change both occurrences of "100" to "100D" in line 10. Same result? Yes - but watch the next two lines:

```
.6799999999999999        68                        68
.6899999999999999        68.99999999999999         69
```

The INT function still doesn't work as expected because all intrinsic functions on the TRS-80 are limited to single precision accuracy. Thus the INT function, in this case, "rounds up" (which is something the BASIC reference manual says will never happen)!

As you may have guessed, this problem can give strange results in computations involving decimals, especially dollars and cents. Why the problem? Because certain numbers that can be represented exactly in decimal (such as .01) cannot be exactly represented in binary, or else the number of binary digits (bits) required for exact binary representation is more than what is available for storage of the number. It would take a lot more space to give a better explanation (see pages 20-25 of TRS-80 ROM ROUTINES DOCUMENTED for a more complete discussion of how numbers are internally stored in the memory of the TRS-80).

How do you avoid this problem, if you are doing computations that require extreme accuracy? Here's a few tips:

1) If you know how to program in assembly language, you could try writing your own routines. Some microprocessors (including the Z-80 used in the TRS-80) will support Binary-Coded Decimal (BCD) arithmetic. The tradeoff here is that more memory is required to support a given precision number. For instance, a single precision number of six digits can be stored in binary in four bytes. "So what?", you might say, since six digits can be stored in three bytes in BCD. But, those three bytes would not include information as to the sign of the number, the location of the decimal point, etc., nor would it allow the overflow into scientific notation allowed by the four-byte binary scheme. Nevertheless, one could wish for the inclusion of BCD arithmetic in a future release of BASIC, since accuracy is required in many applications, and the cost in terms of memory would not be that great (perhaps as little as one or two bytes more per variable).

2) Do all computations using integers, and carry all variables as integers. Note that I do not mean integer precision, but rather numbers of any precision that do not have a fractional part (no digits to the right of the decimal point). The reason this works is that any whole number can be represented exactly in binary - only fractional parts will get us into trouble. As an example: Suppose we are working with dollars and cents. We could use a statement such as this:

```
10 INPUT "ENTER AMOUNT";X: X=INT(X*100+.1)
```

If the user entered, say, 1.49, it would be carried in the computer as 149 (think of it as 149 cents). Any computations done on this number would be accurate as long as whole numbers are used. The ".1" in the above line is a "fudge factor", it's there to make sure that the INT function returns the proper value. When the time comes to print the result (and ONLY then), we use a statement of the form:

```
1000 PRINT USING "$###.##"; X/100
```

Note that numbers stored on disk, etc. should still be in their "multiplied by 100" format.

3) Here are some tricks to "clean up" numbers that may not be stored accurately (for example, when printing the variable returns the proper amount but printing the INT(variable) does not):

A. X=VAL(STR$(X)) (converts to a string & back)

B. X=INT(X*100+.1)/100 (works for dollars and cents, or change "100" to desired value as required)

C. X#=.01D or X#=.01# (always indicate a constant as double precision if it has a fractional part and is being used with double precision variables)

4) Don't expect exact results from the intrinsic functions (exponents, square roots, logarithms, trigonometric functions, etc.). If you need high precision (or if using double precision variables and you want double precision results), you must write your own routines in BASIC or machine language to perform these functions. An article in the August, 1979 issue of CREATIVE COMPUTING tells how to write these functions in BASIC for extended precision.

Now that you know about this problem, go fix that bookkeeping program you wrote, that always seems to be a few cents off in its results (and you never knew why)...

SETDATE - THE MODEL 4 VERSION

by Jack Decker

Many TRS-80 computer users have given up setting the date when they power up their system, and no, they haven't bought one of those battery-powered clock/calendar boards. Instead, they use SETDATE/CMD, a program for the TRS-80 Models I and III, which first appeared in NORTHERN BYTES Volume 5 Number 2 (with bugs, unfortunately, which were corrected in later versions) and on TAS Public Domain Library Disk #001.

Unfortunately, the original version of SETDATE did not work in the Model 4 mode. So, here is SETDATE 6.0, a program which works under both TRSDOS 6 and DOSPLUS 6. Please refer to the original SETDATE article and to the comments in the source code listing below for further information on the operation of SETDATE.

In order for SETDATE to be useful, the DATE prompt that you get at boot-up time must be disabled. Under TRSDOS 6, this is done by issuing a command of the form:

```
SYSTEM (DATE=NO)
```

Unfortunately, this has an undesirable side effect. You see, when you boot TRSDOS 6 with the DATE prompt is enabled [SYSTEM (DATE=YES)] <u>and</u> there is already a valid date in memory, the user is not prompted to set the date, and the date already in memory is used. BUT, if you have the DATE prompt disabled [SYSTEM (DATE=NO)] and you reboot the computer, TRSDOS 6 will (apparently in a fit of spite because you bypassed the DATE routine) erase the date from memory, even if it's valid! SETDATE is written in such a way that if it finds an already valid date in memory when it is executed, it will simply exit to DOS without prompting the user to reset the date again (unless the ! argument is used). But, because TRSDOS 6 has already deleted the date from memory (a "feature" I do not particularly appreciate), SETDATE will never find that already valid date after a reboot. Obviously, a patch to TRSDOS 6 is needed, to keep it from erasing the date in memory at reboot time.

So, here's the patch. You can apply the three patch lines directly from the keyboard, or you can put the following lines into a patch file and execute using the command: DO SETDATE/FIX

```
. SETDATE/FIX - 08/19/85 by Jack Decker
. (with a little help from Hardin Brothers)
.
. Optional patch to SYS0/SYS on TRSDOS 6.2
.
. This patch prevents SYS0/SYS from erasing a valid
. date from memory after a system reboot, when the
. SYSTEM (DATE=NO) command has been used.  Note that
. this patch may also permit an invalid date to
. remain in memory after a reboot.
.
PATCH SYS0/SYS.LSIDOS (D0D,2C=00:F0D,2C=36)
PATCH SYS0/SYS.LSIDOS (D0D,30=00:F0D,30=36)
PATCH SYS0/SYS.LSIDOS (D0D,34=00:F0D,34=36)
.
. End of patch
```

Thanks to Hardin Brothers for passing along the necessary information for me to create this patch. You might be interested in a few of Hardin's comments:

"A pseudo-code [for the operation of the date-setting code in TRSDOS 6.2] works something like this:

```
Collect the current contents of the date area and erase any
        garbage that may be there [this erasure is what
        the above patch eliminates].
If date-prompt = NO, jump to time prompt.
Store current date contents in the input buffer.
Jump to date-parsing routine.
Input date from user.
Parse input. If not valid date, jump back to input.
Store valid date in date area and on screen.
```

"Now, the problem with inhibiting the erasure is that it would be inhibited always, including the first IPL of a session, and so any random garbage would be accepted as a valid date..."

What this all means is that if you're NOT using SETDATE, you probably won't want to apply the above patches to your copy of TRSDOS 6. But there is no problem if you do use SETDATE, because it contains its own check for an invalid date in memory, and will ask you to reset the date if it finds one.

The first time you execute the Model 4 version of SETDATE, you may get a "File Already Open" error message. If that happens, go to DOS READY and type:

```
RESET SETDATE/CMD
```

Once you've done this, you should have no further problems.

Even though the name of the source code is SETDATE6/ASM the program must still be assembled using the filenam SETDATE/CMD. If you use any other filename, it won't work! The normal use of this program is as part of the AUTO command, to have it execute automatically when you boot up your system (see the source code comments for more information).

Those of you that are long-time readers of NORTHERN BYTES will recall that the original SETDATE/CMD program has gone through several permutations. This means two things to you: First, check future issues of NORTHERN BYTES (if any) for bug reports or code improvements. Second, perhaps someday I may convert other programs of the SETDATE series to work under TRSDOS 6. I have already attempted to convert SD/CMD (that's the one that also writes the date to a text file called DATE/TXT, which can then be imbedded into your word-processor files so that your form letters are always printed using today's date). Unfortunately, I have so far been unsuccessful in the conversion, because I keep getting an "Illegal logical file number" error (!) and I haven't yet figured out why (if any of you TRSDOS 6 hackers would like to examine my source code and tell me what I'm doing wrong, I'd be happy to pass along a copy).

As for the overseas versions that write the date backwards (a relative term in this case), those may be a bit more difficult. It turns out that there is an "international" version of TRSDOS 6.2 that stores the date internally in a different format, and I don't have a copy of that version (and you can't get it here in the U.S.A., obviously). So, without a copy of the "international" version to run tests on, I really can't convert the "international" versions of SETDATE very well. Perhaps one of our overseas readers can offer some assistance here. In case you're wondering, here's one way you can tell which version you have: With SYS0/SYS resident in memory, if the byte at 1EEAH is 1BH (a DEC DE instruction) it's the U.S. version, but if it's 13H (INC DE) it's the "international" version.

The source code follows, and again, please remember that it MUST be assembled using the filename SETDATE/CMD!!!

```
00100 ;xxxxx  SETDATE6/ASM  -  Copyright 1985 by Jack Decker
00110 ;Version 6.0 for the Model 4 - creation date 7/6/85.
00120
00130 ;This program eliminates the need to type in the date
00140 ;manually each time you power-up your computer (assuming
00150 ;your DOS can be forced to bypass the "DATE" prompt when
00160 ;it is booted up).
00170
00180 ;This program MUST be assembled using the filename
00190 ;"SETDATE/CMD".  Normally, it is used by placing the
00200 ;filename as part of a DOS "AUTO" command.  You may AUTO
00210 ;a second program along with SETDATE by placing filename
00220 ;of the second program in same AUTO command line,
00230 ;immediately following the SETDATE command.  For example:
00240
00250 ;                   SETDATE PROGRAM2
00260
00270 ;would first execute SETDATE, then PROGRAM2.
00280
00290 ;If there appears to be a valid date already stored in
00300 ;memory (as might happen if the DOS is re-booted without
00310 ;first turning off the power to the system), SETDATE will
00320 ;simply clear the screen and exit.  However, you can
00330 ;force SETDATE to execute by placing a space and
00340 ;exclamation point immediately following the filename:
00350
00360 ;                   SETDATE !
00370 ;         or        SETDATE ! PROGRAM2
00380
00390 ;When SETDATE executes, it will display the date used the
00400 ;last time SETDATE was executed.  You may advance or
00410 ;backspace the date using the arrow keys.  When the
00420 ;correct date is displayed, simply press the ENTER key,
00430 ;which will store the date in memory and also within the
00440 ;SETDATE program itself, for use the next time SETDATE
00450 ;is executed.
00460
00470 ;Questions or comments MUST be accompanied by a self-
```

```
              00480 ;addressed stamped envelope if you live in the U.S.A. or
              00490 ;Canada (Canadian postage is O.K.) and wish a reply.
              00500
              00510 ;                 Jack Decker
              00520 ;                 1804 West 18th Street   Lot # 155
              00530 ;                 Sault Ste. Marie, Michigan  49783
              00540
              00550
3000          00560        ORG    3000H          ;MUST end with 00H
              00570
              00580 ;String and date storage area used by program
              00590
3000 2030     00600 TABLE  DEFW   SUN            ;Table of string pointers
3002 3330     00610        DEFW   MON            ;  point to strings
3004 3930     00620        DEFW   TUE            ;  containing days of
3006 4030     00630        DEFW   WED            ;  the week
3008 4930     00640        DEFW   THU
300A 5130     00650        DEFW   FRI
300C 5730     00660        DEFW   SAT
300E 5F30     00670        DEFW   JAN            ;Table of string pointers
3010 6630     00680        DEFW   FEB            ;  point to strings
3012 6E30     00690        DEFW   MAR            ;  containing months of
3014 7330     00700        DEFW   APR            ;  the year
3016 7830     00710        DEFW   MAY
3018 7B30     00720        DEFW   JUN
301A 7F30     00730        DEFW   JUL
301C 8330     00740        DEFW   AUG
301E 8930     00750        DEFW   SEP
3020 9230     00760        DEFW   OCT
3022 9930     00770        DEFW   NOV
3024 A130     00780        DEFW   DEC
3026 FE       00790 MARKER DEFB   0FEH           ;Start date storage area
3027 55       00800 DATSTR DEFB   85D            ;Year storage
3028 01       00810        DEFB   1D             ;Day storage
3029 07       00820        DEFB   7D             ;Month storage
302A 01       00830        DEFB   1D             ;Day of week storage
302B 6C07     00840 CNTURY DEFW   1900D          ;Century storage
302D 53       00850 SUN    DEFM   'Sunda'        ;Strings containing days
     75 6E 64 61
3032 F9       00860        DEFB   'y'+80H        ;  of week
3033 4D       00870 MON    DEFM   'Monda'
     6F 6E 64 61
3038 F9       00880        DEFB   'y'+80H
3039 54       00890 TUE    DEFM   'Tuesda'
     75 65 73 64 61
303F F9       00900        DEFB   'y'+80H
3040 57       00910 WED    DEFM   'Wednesda'
     65 64 6E 65 73 64 61
3048 F9       00920        DEFB   'y'+80H
3049 54       00930 THU    DEFM   'Thursda'
     68 75 72 73 64 61
3050 F9       00940        DEFB   'y'+80H
3051 46       00950 FRI    DEFM   'Frida'
     72 69 64 61
3056 F9       00960        DEFB   'y'+80H
3057 53       00970 SAT    DEFM   'Saturda'
     61 74 75 72 64 61
305E F9       00980        DEFB   'y'+80H
305F 4A       00990 JAN    DEFM   'Januar'       ;Strings containing
     61 6E 75 61 72
3065 F9       01000        DEFB   'y'+80H        ;  months of year
3066 46       01010 FEB    DEFM   'Februar'
     65 62 72 75 61 72
306D F9       01020        DEFB   'y'+80H
306E 4D       01030 MAR    DEFM   'Marc'
     61 72 63
3072 E8       01040        DEFB   'h'+80H
3073 41       01050 APR    DEFM   'Apri'
     70 72 69
3077 EC       01060        DEFB   'l'+80H
3078 4D       01070 MAY    DEFM   'Ma'
     61
307A F9       01080        DEFB   'y'+80H
307B 4A       01090 JUN    DEFM   'Jun'
     75 6E
307E E5       01100        DEFB   'e'+80H
307F 4A       01110 JUL    DEFM   'Jul'
     75 6C
3082 F9       01120        DEFB   'y'+80H
3083 41       01130 AUG    DEFM   'Augus'
     75 67 75 73
3088 F4       01140        DEFB   't'+80H
3089 53       01150 SEP    DEFM   'Septembe'
     65 70 74 65 6D 62 65
3091 F2       01160        DEFB   'r'+80H
3092 4F       01170 OCT    DEFM   'Octobe'
     63 74 6F 62 65
3098 F2       01180        DEFB   'r'+80H
3099 4E       01190 NOV    DEFM   'Novembe'
     6F 76 65 6D 62 65
30A0 F2       01200        DEFB   'r'+80H
30A1 44       01210 DEC    DEFM   'Decembe'
     65 63 65 6D 62 65
30A8 F2       01220        DEFB   'r'+80H
30A9 1C       01230 MSG    DEFB   1CH                 ;1CH will home cursor
30AA 1F       01240        DEFB   1FH                 ;1FH will clear screen
30AB 0F       01250        DEFB   0FH                 ;0FH turns off cursor
30AC 1A       01260        DEFB   1AH                 ;1AH moves down one line
30AD 10       01270        DEFB   10H                 ;10H activates rev. video
30AE 50       01280        DEFM   'Please set correct date!'
     6C 65 61 73 65 20 73 65 74 20 63 6F 72 72 65 63
     74 20 64 61 74 65 3A
30C6 11       01290        DEFB   11H                 ;11H deselects rev. video
30C7 0D       01300        DEFB   0DH
30C8 1A       01310        DEFB   1AH
30C9 1A       01320        DEFB   1AH
30CA 1A       01330        DEFB   1AH
30CB 50       01340        DEFM   'Please press one of the following keys:'
     6C 65 61 73 65 20 70 72 65 73 73 20 6F 6E 65 20
     6F 66 20 74 68 65 20 66 6F 6C 6C 6F 77 69 6E 67
     20 6B 65 79 73 3A
30F2 0D       01350        DEFB   0DH
30F3 3C       01360        DEFM   '<ENTER> if date is correct'
     45 4E 54 45 52 3E 20 69 66 20 64 61 74 65 20 69
     73 20 63 6F 72 72 65 63 74
310D 0D       01370        DEFB   0DH
310E 3C       01380        DEFM   '<BREAK> to exit without setting date'
     42 52 45 41 4B 3E 20 74 6F 20 65 78 69 74 20 77
     69 74 68 6F 75 74 20 73 65 74 74 69 6E 67 20 64
     61 74 65
3132 0D       01390        DEFB   0DH
3133 3C       01400        DEFM   '<up-arrow> or <right-arrow> to advance
date'
     75 70 2D 61 72 72 6F 77 3E 20 6F 72 20 3C 72 69
     67 68 74 2D 61 72 72 6F 77 3E 20 74 6F 20 61 64
     76 61 6E 63 65 20 64 61 74 65
315E 0D       01410        DEFB   0DH
315F 3C       01420        DEFM   '<down-arrow> or <left-arrow> to backup
date'
     64 6F 77 6E 2D 61 72 72 6F 77 3E 20 6F 72 20 3C
     6C 65 66 74 2D 61 72 72 6F 77 3E 20 74 6F 20 62
     61 63 6B 75 70 20 64 61 74 65
318A 0D       01430        DEFB   0DH
318B 3C       01440        DEFM   '<SHIFT> plus arrow key to change date at
high speed'
     53 48 49 46 54 3E 20 70 6C 75 73 20 61 72 72 6F
     77 20 6B 65 79 20 74 6F 20 63 68 61 6E 67 65 20
     64 61 74 65 20 61 74 20 68 69 67 68 20 73 70 65
     65 64
31BE 0D       01450        DEFB   0DH
31BF 1A       01460        DEFB   1AH
31C0 1A       01470        DEFB   1AH
31C1 1A       01480        DEFB   1AH
31C2 1A       01490        DEFB   1AH
31C3 1A       01500        DEFB   1AH
31C4 53       01510        DEFM   'SETDATE version 6.0 is a public domain
program by Jack Decker.'
     45 54 44 41 54 45 20 76 65 72 73 69 6F 6E 20 36
     2E 30 20 69 73 20 61 20 70 75 62 6C 69 63 20 64
     6F 6D 61 69 6E 20 70 72 6F 67 72 61 6D 20 62 79
     20 4A 61 63 6B 20 44 65 63 6B 65 72 2E
3202 0D       01520        DEFB   0DH
3203 50       01530        DEFM   'Placed in the public domain courtesy of
NORTHERN BYTES, the newsletter for'
     6C 61 63 65 64 20 69 6E 20 74 68 65 20 70 75 62
     6C 69 63 20 64 6F 6D 61 69 6E 20 63 6F 75 72 74
     65 73 79 20 6F 66 20 4E 4F 52 54 48 45 52 4E 20
     42 59 54 45 53 2C 20 74 68 65 20 6E 65 77 73 6C
     65 74 74 65 72 20 66 6F 72
324D 0D       01540        DEFB   0DH
324E 6F       01550        DEFM   'owners and users of the TRS-80 Models I,
III, 4 and 4P.'
```

```
      77 6E 65 72 73 20 61 6E 64 20 75 73 65 72 73 20
      6F 66 20 74 68 65 20 54 52 53 2D 38 30 20 4D 6F
      64 65 6C 73 20 49 2C 20 49 49 49 2C 20 34 20 61
      6E 64 20 34 50 2E
3285 0D        01560        DEFB   0DH
3286 1A        01570        DEFB   1AH
3287 46        01580        DEFM   'For information on how you may obtain
current or back issues of NORTHERN BYTES,'
      6F 72 20 69 6E 66 6F 72 6D 61 74 69 6F 6E 20 6F
      6E 20 68 6F 77 20 79 6F 75 20 6D 61 79 20 6F 62
      74 61 69 6E 20 63 75 72 72 65 6E 74 20 6F 72 20
      62 61 63 6B 20 69 73 73 75 65 73 20 6F 66 20 4E
      4F 52 54 48 45 52 4E 20 42 59 54 45 53 2C
32D6 0D        01590        DEFB   0DH
32D7 70        01600        DEFM   'please contact The Alternate Source, 704
North Pennsylvania Avenue,'
      6C 65 61 73 65 20 63 6F 6E 74 61 63 74 20 54 68
      65 20 41 6C 74 65 72 6E 61 74 65 20 53 6F 75 72
      63 65 2C 20 37 30 34 20 4E 6F 72 74 68 20 50 65
      6E 6E 73 79 6C 76 61 6E 69 61 20 41 76 65 6E 75
      65 2C
331A 0D        01610        DEFB   0DH
331B 4C        01620        DEFM   'Lansing, Michigan 48906 (U.S.A.) or
telephone (517) 482-8270.'
      61 6E 73 69 6E 67 2C 20 4D 69 63 68 69 67 61 6E
      20 34 38 39 30 36 20 28 55 2E 53 2E 41 2E 29 20
      6F 72 20 74 65 6C 65 70 68 6F 6E 65 20 28 35 31
      37 29 20 34 38 32 2D 38 32 37 30 2E
3358 1C        01630        DEFB   1CH              ;1CH will home cursor
3359 10        01640        DEFB   10H              ;Re-enable reverse video
335A 11        01650        DEFB   11H              ;  but keep de-activated
335B 1A        01660        DEFB   1AH              ;1AH moves down one line
335C 1A        01670        DEFB   1AH
335D 9A        01680        DEFB   1AH+80H
335E 1E        01690 ENDYR  DEFB   1EH              ;1EH clears to line end
335F 9D        01700        DEFB   1DH+80H          ;1DH moves to line start
3360 1C        01710 CLS    DEFB   1CH              ;1CH will home cursor
3361 9F        01720        DEFB   1FH+80H          ;1FH will clear screen
3362 44        01730 DATE   DEFM   'DATE'           ;Used to execute DOS DATE
      41 54 45
3366 0D        01740        DEFB   0DH              ;  command
               01750
               01760 ;Start of actual machine-language program
               01770
3367 7E        01780 START  LD     A,(HL)           ;Get argument (!) if any
3368 D621      01790        SUB    '!'              ;Is it exclamation point
336A CCFE34    01800        CALL   Z,SKPSPC         ;Bump HL past excl. point
336D 112836    01810        LD     DE,CMDBFR        ;Temporary command buffer
3370 015000    01820        LD     BC,80D           ;Number of bytes to move
3373 EDB0      01830        LDIR                    ;Move rest of command ln
3375 B7        01840        OR     A                ;Re-test for !
3376 281A      01850        JR     Z,USEPGM         ;Skip date test if ! used
3378 217836    01860        LD     HL,NUMBUF        ;Point to dummy buffer
337B 3E12      01870        LD     A,12H            ;Get date storage area
337D EF        01880        RST    28H
337E 1A        01890        LD     A,(DE)           ;Get year from memory
337F FE64      01900        CP     100D             ;Is it in range 00 - 99?
3381 300F      01910        JR     NC,USEPGM        ;Go if invalid year
3383 13        01920        INC    DE               ;Point to day in memory
3384 1A        01930        LD     A,(DE)           ;Get day from memory
3385 3D        01940        DEC    A                ;Adjust valid to 0 - 30
3386 FE1F      01950        CP     31D              ;Is day 1 - 31 in memory?
3388 3008      01960        JR     NC,USEPGM        ;Go if invalid month
338A 13        01970        INC    DE               ;Point to month in memory
338B 1A        01980        LD     A,(DE)           ;Get month from memory
338C 3D        01990        DEC    A                ;Adjust valid to 0 - 11
338D FE0C      02000        CP     12D              ;Is month 1 - 12 in mem?
338F DA2E34    02010        JP     C,EXIT2          ;Go if memory date valid
3392 21A930    02020 USEPGM LD     HL,MSG           ;Point to message
3395 CD8734    02030        CALL   DSPMSG           ;Display message
3398 012A30    02040 RESTRT LD     BC,DATSTR+3      ;Point to day of wk byte
339B 0A        02050        LD     A,(BC)           ;Get day of week (0 - 6)
339C CDAF34    02060        CALL   PRTSTR           ;Print day of week string
339F CD0434    02070        CALL   PRTCOM           ;Print comma and space
33A2 0B        02080        DEC    BC               ;Point to month byte
33A3 0A        02090        LD     A,(BC)           ;Get month (1 - 12)
33A4 C606      02100        ADD    A,6              ;Offset for string table
33A6 CDAF34    02110        CALL   PRTSTR           ;Print month string
33A9 CD0934    02120        CALL   PRTSPC           ;Print space character
33AC 0B        02130        DEC    BC               ;Point to day byte
33AD 0A        02140        LD     A,(BC)           ;Get day (1 - 31)
```

```
33AE 6F        02150        LD     L,A              ;Put day in L
33AF 2600      02160        LD     H,0              ;HL = day
33B1 C5        02170        PUSH   BC               ;Save date storage pntr
33B2 CDC534    02180        CALL   PRTNUM           ;Print day
33B5 C1        02190        POP    BC               ;Restore date storage ptr
33B6 CD0434    02200        CALL   PRTCOM           ;Print comma and space
33B9 0B        02210        DEC    BC               ;Point to year byte
33BA 0A        02220        LD     A,(BC)           ;Get year (0 - 99)
33BB 4F        02230        LD     C,A              ;Put year in C
33BC 0600      02240        LD     B,0              ;BC = last 2 digits year
33BE 2A2B30    02250        LD     HL,(CNTURY)      ;Get century offset
33BE 2A2B30    02250        LD     HL,(CNTURY)      ;Get century offset
33C1 09        02260        ADD    HL,BC            ;HL = Year (all 4 digits)
33C2 CDC534    02270        CALL   PRTNUM           ;Print year
33C5 215E33    02280        LD     HL,ENDYR         ;Point to ctrl chr string
33C8 CD8734    02290        CALL   DSPMSG           ;Output it to video
33CB 3A0735    02300        LD     A,(CFLAG)        ;Get shifted char flag
33CE B7        02310        OR     A                ;Test for zero byte
33CF 2808      02320        JR     Z,GETKEY         ;If shifted char
33D1 010080    02330        LD     BC,8000H         ;Delay amount
33D4 0B        02340 DLY    DEC    BC               ;Decrement delay count
33D5 78        02350        LD     A,B              ;Reached zero?
33D6 B1        02360        OR     C
33D7 20FB      02370        JR     NZ,DLY           ;If not zero yet
33D9 3E08      02380 GETKEY LD     A,8H             ;Get keystroke if any
33DB EF        02390        RST    28H
33DC 20FB      02400        JR     NZ,GETKEY        ;If no key was pressed
33DE FE80      02410        CP     80H              ;Was it the BREAK key?
33E0 2846      02420        JR     Z,EXIT           ;Exit if BREAK
33E2 FE0D      02430        CP     0DH              ;Was it the ENTER key?
33E4 205B      02440        JR     NZ,NOTCR         ;Go if not ENTER
33E6 217836    02450        LD     HL,NUMBUF        ;Point to dummy buffer
33E9 3E12      02460        LD     A,12H            ;Get date storage area
33EB EF        02470        RST    28H
33EC 212730    02480        LD     HL,DATSTR        ;Point to program date
33EF E5        02490        PUSH   HL               ;Save program date pntr
33F0 010300    02500        LD     BC,3             ;Number of bytes to move
33F3 EDB0      02510        LDIR                    ;Move from program to mem
33F5 0600      02520        LD     B,0              ;Logical Record Lngth=256
33F7 110835    02530        LD     DE,FCB           ;File Control Block pntr
33FA 212835    02540        LD     HL,FILBUF        ;File I/O Buffer pntr
33FD 3E3B      02550        LD     A,3BH            ;DOS OPEN routine
33FF EF        02560        RST    28H
3400 2032      02570        JR     NZ,ERREXT        ;Go if error
3402 3E43      02580        LD     A,43H            ;READ sector 0 to buffer
3404 EF        02590        RST    28H
3405 202D      02600        JR     NZ,ERREXT        ;Go if error
3407 E1        02610        POP    HL               ;Get program date pointer
3408 D5        02620        PUSH   DE               ;Save FCB pointer
3409 115235    02630        LD     DE,MARKER-TABLE+4+FILBUF
               02640 ;Above instruction points DE to first location in sector
               02650 ;zero of disk file that can possibly contain MARKER byte
340C 1A        02660 FNDDAT LD     A,(DE)           ;Get byte frm disk sector
340D 13        02670        INC    DE               ;Bump pointer to next loc
340E FEFE      02680        CP     0FEH             ;MARKER byte found?
3410 20FA      02690        JR     NZ,FNDDAT        ;Try again if not
3412 010400    02700        LD     BC,4             ;Move 4 bytes frm program
3415 EDB0      02710        LDIR                    ;  storage to disk sector
3417 D1        02720        POP    DE               ;Restore FCB pointer
3418 3E44      02730        LD     A,44H            ;Reset to sector zero
341A EF        02740        RST    28H
341B 2017      02750        JR     NZ,ERREXT        ;Go if error
341D 3E49      02760        LD     A,49H            ;Write sector back to dsk
341F EF        02770        RST    28H
3420 2012      02780        JR     NZ,ERREXT        ;Go if error
               02790 ;Do NOT attempt to CLOSE file after above procedure!!
               02800 ;It is not necessary, and will truncate file if you do!
3422 216233    02810        LD     HL,DATE          ;Point to "DATE" string
3425 3E19      02820        LD     A,19H            ;Execute DOS DATE command
3427 EF        02830        RST    28H              ;  to reset day bytes
3428 216033    02840 EXIT   LD     HL,CLS           ;Point to clr screen msg
342B CD8734    02850        CALL   DSPMSG           ;Display message
342E 212836    02860 EXIT2  LD     HL,CMDBFR        ;Point to new command bfr
3431 3E18      02870        LD     A,18H            ;Execute next DOS command
3433 EF        02880        RST    28H              ;  (if any) and exit
3434 F640      02890 ERREXT OR     40H              ;Display normal err msg
3436 F5        02900        PUSH   AF               ;Save error number
3437 0E1F      02910        LD     C,1FH            ;1FH=clr to end of screen
3439 3E02      02920        LD     A,2H             ;Output byte to video
343B EF        02930        RST    28H
343C F1        02940        POP    AF               ;Restore error code
```

```
3430 4F       02950        LD     C,A              ;Error number in C
343E 3E1A     02960        LD     A,1AH            ;Exit to error routine
3440 EF       02970        RST    28H
3441 FE5B     02980 NOTCR  CP     5BH              ;Was keystroke up-arrow?
3443 47       02990        LD     B,A              ;Save keystroke in B
3444 280B     03000        JR     Z,ARROW          ;Go if up-arrow
3446 F610     03010        OR     10H              ;Make shifted=unshifted
3448 FE18     03020        CP     18H              ;Invalid keystroke if
344A 388D     03030        JR     C,GETKEY         ;  less than ASCII 18H
344C FE1C     03040        CP     1CH              ;Keystroke valid if less
344E 3089     03050        JR     NC,GETKEY        ;  than ASCII 1CH
3450 A8       03060        XOR    B                ;Check for shifted char
3451 320735   03070 ARROW  LD     (CFLAG),A        ;Save shifted char flag
3454 78       03080        LD     A,B              ;Restore original char
3455 219833   03090        LD     HL,RESTRT        ;RET addr for following
3458 E5       03100        PUSH   HL               ;Save it on stack
3459 212A30   03110        LD     HL,DATSTR+3      ;Point to day-of-week
345C 0F       03120        RRCA                    ;Shift bit 0 into carry
345D 3030     03130        JR     NC,BACK          ;If key was back/down arr
345F 34       03140 ADV    INC    (HL)             ;Increment day of week
3460 7E       03150        LD     A,(HL)           ;Get day of week
3461 FE07     03160        CP     7                ;Is it greater than 6?
3463 3802     03170        JR     C,SETDAY         ;Go if valid day
3465 3600     03180        LD     (HL),0           ;Else reset to Sunday
3467 2B       03190 SETDAY DEC    HL               ;Point to month storage
3468 7E       03200        LD     A,(HL)           ;Get month
3469 CDE234   03210        CALL   MAXDAY           ;Get number days in month
346C 2B       03220        DEC    HL               ;Point to day storage
346D 7E       03230        LD     A,(HL)           ;Get current day
346E B9       03240        CP     C                ;Compare with maximum
346F 34       03250        INC    (HL)             ;Advance day of month
3470 D8       03260        RET    C                ;Finished if valid day
3471 3601     03270        LD     (HL),1           ;Else first of new month
3473 23       03280        INC    HL               ;Point to month storage
3474 7E       03290        LD     A,(HL)           ;Get current month
3475 FE0C     03300        CP     12D              ;See if it's December
3477 34       03310        INC    (HL)             ;Advance month count
3478 D8       03320        RET    C                ;Finished if valid month
3479 3601     03330        LD     (HL),1           ;Else January of new year
347B 2B       03340        DEC    HL               ;Bump pointer back down
347C 2B       03350        DEC    HL               ;  to year storage
347D 7E       03360        LD     A,(HL)           ;Get current year
347E FE63     03370        CP     99D              ;See if last of century
3480 34       03380        INC    (HL)             ;Advance year count
3481 D8       03390        RET    C                ;Finished if valid year
3482 3600     03400        LD     (HL),0           ;Else reset year count
3484 116400   03410        LD     DE,100D          ;Add 100 years to century
3487 2A2B30   03420 CHGCEN LD     HL,(CNTURY)      ;Get current century
348A 19       03430        ADD    HL,DE            ;Adjust century offset
348B 222B30   03440        LD     (CNTURY),HL      ;  and re-save it
348E C9       03450        RET                     ;Finished for sure
348F 35       03460 BACK   DEC    (HL)             ;Decrement day of week
3490 F29534   03470        JP     P,SETDA2         ;Go if valid day
3493 3606     03480        LD     (HL),6           ;Else reset to Saturday
3495 2B       03490 SETDA2 DEC    HL               ;Point to month storage
3496 7E       03500        LD     A,(HL)           ;Get month
3497 2B       03510        DEC    HL               ;Point to day storage
3498 35       03520        DEC    (HL)             ;Decrement day of month
3499 C0       03530        RET    NZ               ;Finished if valid day
349A 3D       03540        DEC    A                ;Else A=# of previous mth
349B CDE234   03550        CALL   MAXDAY           ;Get # days previous mnth
349E 71       03560        LD     (HL),C           ;Day=last day prev. month
349F 23       03570        INC    HL               ;Point to month storage
34A0 35       03580        DEC    (HL)             ;Decrement month count
34A1 C0       03590        RET    NZ               ;Finished if valid month
34A2 360C     03600        LD     (HL),12D         ;Else December prev. year
34A4 2B       03610        DEC    HL               ;Bump pointer back down
34A5 2B       03620        DEC    HL               ;  to year storage
34A6 35       03630        DEC    (HL)             ;Decrement year count
34A7 F0       03640        RET    P                ;Finished if valid year
34A8 3663     03650        LD     (HL),99D         ;Else last yr prev cntury
34AA 119CFF   03660        LD     DE,-100D         ;Subtract 100 frm century
34AD 18D8     03670        JR     CHGCEN           ;Adjust century & finish
34AF 07       03680 PRTSTR RLCA                    ;A=AX2 (2 byte pointers)
34B0 6F       03690        LD     L,A              ;L=LSB string table addr
34B1 2630     03700        LD     H,TABLE<-8       ;H=MSB string table addr
34B3 7E       03710        LD     A,(HL)           ;A=LSB actual string addr
34B4 23       03720        INC    HL               ;Point to MSB string addr
34B5 66       03730        LD     H,(HL)           ;H=MSB actual string addr
34B6 6F       03740        LD     L,A              ;HL=string location addr
34B7 7E       03750 DSPMSG LD     A,(HL)           ;Get byte to display
34B8 B7       03760        OR     A                ;See if zero terminator
34B9 C8       03770        RET    Z                ;Finished if zero byte
34BA F5       03780        PUSH   AF               ;Save sign flag status
34BB E67F     03790        AND    7FH              ;Mask off bit 7
34BD CDDB34   03800        CALL   VDSP             ;Display it on video
34C0 F1       03810        POP    AF               ;Restore sign flag
34C1 F8       03820        RET    M                ;Finished if bit 7 set
34C2 23       03830        INC    HL               ;Advance string pointer
34C3 18F2     03840        JR     DSPMSG           ;Go print next byte
34C5 117836   03850 PRTNUM LD     DE,NUMBUF        ;Point to number buffer
34C8 D5       03860        PUSH   DE               ;Save number buffer ptr
34C9 3E61     03870        LD     A,61H            ;Convert number in HL to
34CB EF       03880        RST    28H              ;  ASCII
34CC AF       03890        XOR    A                ;Zero byte in A register
34CD 12       03900        LD     (DE),A           ;Mark end of number
34CE E1       03910        POP    HL               ;Restore buffer pointer
34CF CDFE34   03920        CALL   SKPSPC           ;Skip leading space chrs
34D2 18E3     03930        JR     DSPMSG           ;Display converted number
34D4 3E2C     03940 PRTCOM LD     A,','            ;Comma character in A
34D6 CDDB34   03950        CALL   VDSP             ;Display it on video
34D9 3E20     03960 PRTSPC LD     A,' '            ;Space character in A
34DB C5       03970 VDSP   PUSH   BC               ;Save BC
34DC 4F       03980        LD     C,A              ;Byte to display in C
34DD 3E02     03990        LD     A,2H             ;Output byte to video
34DF EF       04000        RST    28H
34E0 C1       04010        POP    BC               ;Restore BC
34E1 C9       04020        RET
34E2 0E1E     04030 MAXDAY LD     C,30D            ;C=# days in some months
34E4 FE04     04040        CP     4                ;Is month April?
34E6 C8       04050        RET    Z                ;Return if April
34E7 FE06     04060        CP     6                ;Is month June?
34E9 C8       04070        RET    Z                ;Return if June
34EA FE09     04080        CP     9                ;Is month September?
34EC C8       04090        RET    Z                ;Return if September
34ED FE0B     04100        CP     11D              ;Is month November?
34EF C8       04110        RET    Z                ;Return if November
34F0 0C       04120        INC    C                ;C=31 (# days most mnths)
34F1 FE02     04130        CP     2                ;Is month February?
34F3 C0       04140        RET    NZ               ;Return if 31 day month
34F4 0E1C     04150        LD     C,28D            ;C=# days in February
34F6 3A2730   04160        LD     A,(DATSTR)       ;Get current year
34F9 E603     04170        AND    3                ;Year divisible by 4?
34FB C0       04180        RET    NZ               ;If not leap year
34FC 0C       04190        INC    C                ;C=29 (# days leap Feb.)
34FD C9       04200        RET                     ;Finished
34FE F5       04210 SKPSPC PUSH   AF               ;Save AF
34FF 23       04220 SKPSP2 INC    HL               ;Bump to next character
3500 7E       04230        LD     A,(HL)           ;Get character
3501 FE20     04240        CP     20H              ;Is character a space?
3503 28FA     04250        JR     Z,SKPSP2         ;Skip space characters
3505 F1       04260        POP    AF               ;Restore AF
3506 C9       04270        RET
3507 00       04280 CFLAG  DEFB   0                ;Shifted character flag
3508 53       04290 FCB    DEFM   'SETDATE/CMD'    ;File Control Block area
     45 54 44 41 54 45 2F 43 4D 44
3513 03       04300        DEFB   3                ;  with program filename
0014          04310        DEFS   20D              ;  (total 32 bytes)
0100          04320 FILBUF DEFS   100H             ;File I/O buffer area
0050          04330 CMDBFR DEFS   80D              ;Store remaining DOS cmds
0008          04340 NUMBUF DEFS   8                ;Scratch number storage
3367          04350        END    START
00000 TOTAL ERRORS

ADV     345F   APR     3073   ARROW   3451   AUG     3083   BACK    348F
CFLAG   3507   CHGCEN  3487   CLS     3360   CMDBFR  3628   CNTURY  302B
DATE    3362   DATSTR  3027   DEC     30A1   DLY     33D4   DSPMSG  34B7
ENDYR   335E   ERREXT  3434   EXIT    3428   EXIT2   342E   FCB     3508
FEB     3066   FILBUF  3528   FNDDAT  340C   FRI     3051   GETKEY  33D9
JAN     305F   JUL     307F   JUN     307B   MAR     306E   MARKER  3026
MAXDAY  34E2   MAY     3078   MON     3033   MSG     30A9   NOTCR   3441
NOV     3099   NUMBUF  3678   OCT     3092   PRTCOM  34D4   PRTNUM  34C5
PRTSPC  34D9   PRTSTR  34AF   RESTRT  3398   SAT     3057   SEP     3089
SETDA2  3495   SETDAY  3467   SKPSP2  34FF   SKPSPC  34FE   START   3367
SUN     302D   TABLE   3000   THU     3049   TUE     3039   USEPGM  3392
VDSP    34DB   WED     3040
```

Coming Next Issue:

Adding External Drives to the Model 4P!

TANDON DRIVE DOOR HINGE REPLACEMENTS

[Reprinted from Orange Bytes, the magazine of the North Orange County (California) Computer Club. Please note the disclaimer at the end of this article.]

A piece of equipment is only as strong as its weakest link. In the case of the Tandon drive [found in many TRS-80 computer systems, as well as in many other brands of computer] the weak link may well be the hinge that the disk door pivots on. If you have had a failure of this simple component you know the result - since the door operates the drive mechanism itself your computer is out of business until you make repairs. It happened to us, and when we called our Kaypro dealer about it he told us that it would cost about $40 to fix, and the same trouble prone part would be used in replacement. Since we are in the business of making parts for the aviation industry we decided to see if we couldn't come up with something better ourselves. Using aircraft grade heat treated aluminum alloy we fabricated a new hinge that absolutely cannot fail the way the stock plastic part does - in fact it should outlast the rest of the computer. The hinges worked out so well that we have made more for other Kaypro [and presumably other Tandon drive -editor] users, and now offer the parts to whoever would like a set.

If you would like to order a pair of hinges send $10 cash, check, or postal money order (cash or money order gets the quickest response) to:

Steve Govus
Route 2 Box 683
Shipshewana, Indiana 46565

Complete installation instructions are included. It takes only about half an hour to install a set, requires no special tools, and anyone who can run a screwdriver has the necessary mechanical background.

[ORANGE BYTES Editor's Note: This article is offered not as an advertisement but as a public service for the systems that it makes reference to.

Products, services, etc., mentioned or offered within the pages of the Orange bytes have not been screened or verified by the North Orange County Computer Club, its staff or officers. Persons wishing to purchase or use any item, service, or product mentioned or published within the pages of the Bytes do so at their own risk.

Also, any person who would send cash for the purchase of a mailorder item (in or out of state) might be interested in some great ocean-front property that I have for sale in Kansas.]

[NORTHERN BYTES Editor's Note: We echo the above disclaimer, since we have had no dealings whatsoever with this firm and therefore cannot vouch for them in any way. But if you hate water, I have some great desert property in Michigan's Upper Peninsula that you might prefer to the Kansas ocean-front land...]

ANOTHER TRSDOS 1.3 PATCH

by Andy Levinson

[The following is excerpted from an article in The Interface newsletter of the San Gabriel Valley TRS-80 Users Group:]

One of the problems still in TRSDOS 1.3 is a wonderful little bug that can destroy the operating system environment. The bug is particularly ironic because it results from someone's efforts to keep the directory GAT sector looking clean even though TRSDOS 1.3 (by itself) does not permit a user to view that sector.

The bug is in the AUTO command. If you have ever looked at the GAT sector on a TRSDOS 1.3 disk, you may have noticed that the GAT can only hold a 32-character AUTO command (actually only 31 since the command must end with a carriage return). My official TRSDOS 1.3 manual makes no mention of the limitation. No big deal because who enters AUTO commands that long?

But what if you did? The wonderful charm of computers and programming is that you can always depend on someone at some time to do something out of the ordinary. Test it yourself with this command:

```
AUTO GREAT WAY TO CRASH A TRSDOS DISK
```

Surprise! For a tidy GAT sector, AUTO pads the command buffer with up to 32 spaces. What if there are more than 32 characters in the AUTO command argument? Simple math states that 32 minus 32 equals 0 and any machine language student knows that 0 invariably means 256 on an 8-bit computer. That means pad out the 64 character keyboard buffer with 256 spaces. Oops! There went a page of operating system code (perhaps this idea came from the same programmer who decided to keep directory sectors tidy by clearing the entire file entry, thereby precluding recovery, whenever a file is killed.)

Anyway, here is a simple TRSDOS 1.3 patch that ends the problem I just described:

```
PATCH *6 (ADD=531A,FIND=3620,CHG=0000)
```

I didn't spend the time for a complete solution. This patch merely eliminates padding. Disk zappers can now see trailing junk in the GAT sector after a "short" AUTO command. But an unduly long AUTO command will only display garbage on the screen. It will not crash the system.

MODEL I/III SELECTIVE RESTORE

Ever wished you could RESTORE to something other than the first DATA statement? Let's suppose, for example, that you want to be able to RESTORE to the 51st DATA item of a list. Try this: At the beginning of your program, insert this line:

```
FOR X=1 TO 50: READ X$: NEXT: X=PEEK(16639): Y=PEEK(16640):
                          RESTORE
```

Now, whenever you wish to RESTORE to the 51st DATA item, simply execute the statements POKE 16639,X: POKE 16640,Y . Of course, X and Y can be any valid variable names, but remember not to use them elsewhere in your program.

What we're doing, of course, is setting the DATA pointer to read the 51st DATA item, then "remembering" the position of the pointer by saving it in the two variables (X and Y in this case). Later, we can simply POKE the pointer position back into place, to reset the pointer to the 51st DATA item (without the necessity of doing dummy READs to get to where we want to be). By using this technique, you can actually subdivide a DATA list into several segments, and get back to the start of any one of them by doing the proper POKEs.

AN ELECTRONICS INVASION FROM 'DOWNUNDA'

Perhaps you've seen mentions of Dick Smith Electronics here in NORTHERN BYTES, usually in articles that originated in Australia. If not, you've surely seem mentions of the SYSTEM-80 computer, which was the TRS-80 clone that Dick Smith sold in Australia for a few years (the same computer was sold here as the PMC-80).

Well, watch out, America, Dick Smith Electronics has opened its first U.S. stores in California, with more on the way. And their catalog is something to behold, it's reminiscent of the catalogs that companies like Olsen Electronics and Lafayette Radio used to send out. It's a very readable catalog, with lots of information and even bits of humor sprinkled throughout (there's also a picture of Dick's smiling face on nearly every page, but what the heck...). There's a 23 page section of technical information, that contains everything from resistor and capacitor color codes and a fairly comprehensive section of semiconductor and IC data, to antenna data, a decibel chart, a section on suppressing automotive electrical noise, and several other handy reference charts. And, in case you run into an Australian term you don't understand, there's a section entitled "Learn to Speak Strine" to help you translate.

The catalog bears a cover price of $2.00 (and is worth every penny of it just for the technical section), but I got mine for free just by calling their toll free number, (800) 332-5373, and requesting a copy. AND you get two one dollar coupons, each redeemable against any purchase of five dollars or more.

Unfortunately, there's not much in the way of computer equipment here, and some of the prices seem just a shade on the high side (actually, they're probably comparable to Radio Shack's prices on many items). But the catalog is fun to read, and they have a lot of electronic goodies I haven't seen elsewhere. If you (or your kids) have any interest in electronics at all, be sure to get a copy of this catalog. If you don't want to call, the address is: Dick Smith Electronics, P.O. Box 8021, Redwood City, California 94063. Besides the toll free number mentioned above, they have a regular phone: (415) 368-1066, a TELEX number: 160488 DICKS USA, and a facsimile machine: (415) 368-0140.

By the way, Dick Smith Electronics is expanding in the U.S., and they want to employ electronics enthusiasts. Positions are available now in California, and will be opening up in other areas in the future. Write to their Personnel Manager at P.O. Box 2249, Redwood City, California 94043.

Diskette Shopping Made Simple

Use the handy order form to order diskettes in the quantities desired. Use of the product codes to the left of the product helps insure that your order is processed properly. All diskettes are soft-sectored. "Flipples" are punched for use on both sides with a single headed disk drive. All DSDD (double sided, double density) media will work on both single and double headed disk drives. All media is fully guaranteed for five years from the date of purchase.

DATE: ____________________
NAME: ____________________
ADDRESS: ____________________
____
TOTAL AMOUNT PURCHASED: $__________
VISA ______ MASTERCARD ______ CHECK ______ OTHER ______
CARD NUMBER: ____________________
EXPIRATION DATE: __________
SPECIAL INSTRUCTIONS: ____________________
____

We also accept pro forma invoices and properly signed purchase Orders from most schools and governmental agencies.

Package of 10 diskettes with Boxes, Labels, Sleeves and Write Protect Tabs:

Item Code	Price	Style	QTY.
SSDDX	$9.95	SSDD	
DSDDX	$11.95	DSDD	
MDSFLX	$12.95	Pre-punched DSDD Flipples	
MDSMUX	$12.95	Multi-Colored DSDD	

Package of 25 BULK diskettes W/Labels, Sleeves and Write Protect Tabs (you save us the labor of putting them together; YOU do it as you need the diskettes):

Item Code	Price	Style	QTY.
MSSD25	$19.95	SSDD	
MDSD25	$22.95	DSDD	
MDSF25	$24.95	Pre-punched DSDD Flipples	
MDSM25	$24.95	Multi-Colored DSDD	

Package of 100 diskettes with Labels, Sleeves and Write Protect Tabs:

Item Code	Price First 100	Price Extra 100's	Style	FIRST	EXTRA
MSSDDC	$79.95		SSDD		
MSSDDD		$74.95	SSDD		
MDSDDC	$89.95		DSDD		
MDSDDD		$84.95	DSDD		
MDSFLC	$95.95		"FLIPPY"		
MDSFLD		$89.95	"FLIPPY"		
MDSMUC	$95.95		COLOR		
MDSMUD		$89.95	COLOR		

Package of 100 BULK diskettes with Labels, Sleeves and Write Protect Tabs:

Item Code	Price First 100	Price Extra 100's	Style	FIRST	EXTRA
MSSDBC	$75.95		SSDD		
MSSDBD		$69.95	SSDD		
MDSDBC	$85.95		DSDD		
MDSDBD		$79.95	DSDD		
MDSDFC	$90.95		FLIPPY		
MDSDFD		$84.95	FLIPPY		
MDSDMC	$90.95		COLOR		
MDSDMD		$84.95	COLOR		

VISA

These prices are CASH prices and reflect a 4% cash discount. You pay the first $3 in shipping; we pay the rest.

MasterCard

Phone: (517) 482-8270

The Alternate Source

704 North Pennsylvania Avenue
Lansing, Michigan 48906

NORTHERN BYTES

Subscription Information

Northern Bytes is edited by Jack Decker and published on an irregular basis by The Alternate Source Information Outlet. Back issues are available starting with Volume 5, Number 1. Issues prior to that are not available. Some of the most valuable articles from earlier issues may be reprinted in future issues of Northern Bytes. Currently there are eight back issues available for Volume 5, as well as all issues from Volume 6. All back issues are $2 each.

It is very easy to be placed on the Northern Bytes REGULAR list. Simply place your address, Visa or MasterCard number and expiration date on file with us. We will start with the issue you request. We do not bill you for ANY ISSUE until that issue has been mailed. This way, we can continue to offer you top quality information with absolutely no risk to you. There's no question of what to do about unfulfilled issues if we decide to quit publishing. Unless otherwise requested, we presume your subscription will extend through the month of your expiration date.

Don't have a charge card, huh? We understand the myriad of reasons for not having them and we feel that a "To-Be-Invoiced" policy could help increase the demand for Northern Bytes. If you'll do it for us, we'll do it for you. Would you like to be placed on a regular list TO BE BILLED for each issue? You could then send a check for the issues as they are mailed. If you didn't send a check, we would presume that your interest has died and discontinue your subscription. The only requirement for getting onto the list is to pay for the first issue up front; the next will be mailed automatically, if you request. You are insured that you will receive top of the line TRS-80 information as it is released. Ask to be placed on the NO RISK "To-Be-Invoiced Northern Bytes List".

Call or write, but SIGN UP TODAY!

The Alternate Source Information Outlet
704 North Pennsylvania Avenue
Lansing, MI 48906
(517) 482-8270

NORTHERN BYTES
c/o Jack Decker
1804 West 18th Street
Lot # 155
Sault Ste. Marie, Michigan 49783
MCI Mail Address: 102-7413
Telex: 6501027413
(Answerback: 6501027413 MCI)

Bulk Mail
U.S. Postage Paid
Permit #815
Lansing, MI

POSTMASTER: If undeliverable return to:
The Alternate Source, 704 North Pennsylvania Avenue, Lansing, Michigan 48906

To: