

PORT TO PORTAL -- Editorial

I'm sure you've noticed that this issue is bursting at the seams! The four-page white insert includes **Staunch's** first two ads; my thanks to Clay Montgomery of SigmaSoft and Lee Hart of the "new" TMSI. And you'll also find the first in a series of columns on HDOS 3.0. After some hunting, I've been able to fulfill a promise I made to you when I assumed editing from Hank. This column is written by Rick Streeter and you've already seen blurbs on his HDOS 3.0 enhancements in earlier issues. Welcome, Rick; we students are present and awaiting your tutorial!

You'll also discover the **VENDOR.UPDATE** section **inside** the insert; it just barely fit the one column I had available, right beside the memory maps for Rick's column. But even with the extra space, I had to cut some material I wanted to run.

In fact, that points up the need to appear more frequently, as many of you are asking. I'll announce plans for the new year in the next issue. (I'm now close to doing a mass promotional mailing.) But because **Staunch's** exact publication schedule is still up in the air, **please do not send your renewals till then.** I'm presently **returning all checks** until we know if we increase frequency in January.

You'll see another software list in this issue, too. Studio Computers of Birmingham, MI, operated by Ray Massa, is releasing some of its 8-bit software into the public domain. While supplies last, the HDOS products listed on the insert include the original hardcopy documentation. I'm presently working on more of Ray's contract authors.

Finally, I must apologize for the delay getting this issue to you. But my employer has made heavy demands on my time. Moreover, I lost about two weeks gearing up for, attending, and winding down from this year's CHUGCON. You'll read my report on the conference inside. Further, my thanks to Dave Stoner for proofing much of this issue. And the editorial deadline for #9 is 15 Oct.

-- Kirk L Thompson

THE EIGHT-BIT R/W -- Letters

A patch for HFM by PS:

[From Pat Swayne, HUG Software Engineer, HUG, Box 217, Benton Harbor, MI 49022-0217] In your most recent newsletter, Hank Lotz reported on two bugs in the HUG File Manager (CP/M version, on HUG disk 885-1246[-37]). These bugs are in the Copy and Newdisk commands of the program. I have found the fixes for both of them. To fix the Copy bug, locate these lines in the assembly source code:

```
STA      TMPUSR ; SET TEMP. USER
NOUSER: LXI    D,FCBB
```

Add a line between these lines, so that they look like this:

```
STA      TMPUSR ; SET TEMP. USER
INX      H      ; MOVE POINTER PAST USER NO.
NOUSER: LXI    D,FCBB
```

To fix the bug in the Newdisk command, locate these lines:

```
CALL     CBDOS ; SET NEW USER
NOUSER: MOV  A,M ; GET DRIVE CODE
```

Add a new line, to get:

```
CALL     CBDOS ; SET NEW USER
```

```
INX      H      ; MOVE POINTER PAST USER NO.
NOUSER: MOV  A,M ; GET DRIVE CODE
```

Then re-assemble the program, and the Copy and Newdisk commands will work correctly...

By the way, 8-bit code is not completely dead here at HUG. I am working on a program called MagBase, which is a magazine article database system. An MS-DOS/ZDOS version of MagBase [is] announced in the July REMark [see p. 41], along with a notice that an 8-bit CP/M version is coming. Well, the CP/M version is finished now, except for some final debugging that will not be possible until I have created a large enough database for testing purposes. The MS-DOS and CP/M versions can share files, so a database built up by one can be used by the other.

The reason for MagBase is to create a database of all REMark articles since day one. Articles will be listed by title, author, date, issue and volume, page no., and classification. Versatile search routines will let you quickly find articles you need. The entire REMark magazine database will be sold as a separate product from MagBase. MagBase itself will come with a database of the 1987 year of REMark...

I may do an HDOS version of MagBase. Do you think that there are enough H8 and H89 users that don't have or hate CP/M to justify the effort? [Thanks for the patch, Pat. If anyone has an interest in an HDOS version, write directly to Pat. -Ed.]

[From Hank Lotz, Pittsburgh, PA] I have determined that the SET TEMP. USER line in [Pat's] letter [above] is at line 319 of the source. And the SET NEW USER is at line 786 of the source ... As I [wrote] Pat, the changes worked fine.

I also found out that HFM.ASM takes 2 or 3 reads to get it all into MAGIC WAND. Both of Swayne's edits, however, were accessible at read number 1.... [Thanks for the confirmation, Hank. For more correspondence between PS: and Hank, see this issue's "Miscellany" section. -Ed.]

Calling all Canadians

[From Roger F. Dupuis, 978 Francis, Longueuil, Quebec J4J 1E4] I am looking for the dBASE II program for my dear H-89. Can we find it at a reduced price? Or is there something quite similar (since I use it at work on a "PEE-SEE," both dBASE II and dBASE III). I would like something that the user can program to get specific reports (like home inventory, etc.). I did some programming at work, so I would do the same for my machine. What about FOXbase?

If the readers don't mind, I would like to get in touch with the Canadian people registered to your publication ... We could save some duty fees on software sent back and forth.

I am looking for a program specialized in genealogy ... I understand that ROOTS/M was the one ... [Are there others] or does it have some upgrade ... or a CP/M version of the PAF (Personal Ancestral File...)? Or do I have to use the "PEE-SEE" at work? I can use either the CP/M or HDOS operating system. [The package I have, Hoyle and Hoyle's QUERY!3, will do the database job, but it's not dBASE compatible. And I vaguely recall a genealogy package called ROOTS/89. Could you readers help Roger out? -Ed.]

Antique or classic?

[From Stephen Kaiser, Cambridge, MA] Let's face it -- if you have an 8-bit machine, you've got to learn to love obsolescence. We're the Homeless in an MS-DOS world. We need an orphanage and you provide one...

There are some people who believe that antiques should be museum pieces, and not ... used for any heavy work. I believe antiques should be functional and actively used...

What will happen to old computers? Will they go from being antiques to being classics? My best guess at this point is that most computers become antiques within 5 years from the end of the model run, compared to 25 years for cars. When does a computer become a classic? It's too early to tell, because the micro business is only a decade old, but I would guess its in the 10-15 year range. This may be the point where our old machines start going up in value, rather than down!

All this is a way of saying how important your efforts are to keep the 8/89 flame alive. Every orphan group started where we are now, and there is no reason to get panicked into the PR man's line that New-Is-Always-Better. As experienced 89 users, we know better! [Stephen, your discussion reminds me of the old-instrument concerts that the Library of Congress gives from time to time. To keep those old Guarneri and Stradivari in fine fiddle, they have to be played regularly! -Ed.]

SUPERSET character sets

[From Mark Hunt, PHS Native Hospital, Barrow, AK 99723] I've just received Lee Hart's SUPERSET, SUPERFONT, and SUPERCLOCK (H/Z-19/89 alternate [and] extended character sets), and I must say that I am very impressed.

I have created a few alternate character sets for Lee to include in SUPERSET/SUPERFONT: Esperanto, Spanish, and Russian, as well as an additional English font. In addition, I am trying to complete a set for Arabic, and a subset of the International Phonetic Alphabet (broad transcriptions of Esperanto, Spanish, Russian, and Arabic). In a couple of months, you should be able to get these from Lee. Don't know what he will charge.

Perhaps you can help me. I'd like to survey your choices of software; that is, how do you utilize your extra [and] extended SUPER capabilities? Which HDOS [and] CP/M (I use HDOS) text editors [and] processors do you use? What about device drivers [and] printers ... and, have any of you created new downloadable printer fonts? (On disk - SS or DS/HS/40T/SD)

I'll publish the results of this survey in a few months - and, please do not expect me to answer your letters too quickly....

MPI printer driver for HDOS

[From Parks Watson, 4113 W. Franklin St., Richmond, VA 23221] ...When I bought my first printer ... an MPI-99G, from the Virginia Beach HEC store ... they got me an HDOS driver from a member of THUG (Tidewater Heath Users Group), a super ASM programmer. This is a great driver which has the usual SET features but also what he calls "run time setables" or args. to the COPY command which temporarily override the SET parameters; vis., "COPY LP:B4C1DF=FNAME.EXT" will print in serif font (B4), one tab left margin (C1), double spacing (D) and formfeed (F) at end of file. Other arguments are for LPI which automatically sets the appropriate form length, CPI which automatically sets line length, disable driver paging, stop at each new page (CTRL-Q to restart), print only even numbered pages or odd number pages and print unidirectionally. These arguments are effective only for the current printout. And all of this in only six sectors...

Incidentally, this driver is not copyrighted and the author doesn't even give himself a credit line in the documentation ... If any HDOS types with an MPI printer would like a copy, if they will send me an INITed hard sector disk with return postage, I will be glad to copy the driver and the documentation onto it. I don't have the source code for it.

It is for the H8-4/H89 serial interface only and is compatible with all versions of HDOS through 2.0 (50.06.00). [Thanks for the offer, Parks. And I hope to make more solid contact with this world-class device driver programmer from THUG since, when he stopped by my booth at CHUGCON, he offered to send me his entire collection! -Ed.]

WANTED

[From Albert F. Bjorling, Box 216, Circleville, NY 10919, (914) 361-3272]

CP/M SOFTWARE PACKAGE - Looking for a CP/M-80 version of 'QUICK CODE' by Fox and Geller. Wish to use this ... with dBASE II. May be new or used with documentation. Formats I can read directly are: 8 inch IBM, H/Z Z37, Z17, Z47, and Magnolia MMS.

TRACTOR - for ANDERSON-JACOBSON model #AJ832 printer.

H-SCOOP ISSUE #52 (July 84).

Answers to questions

[From Don Rohde, Aurora, IL] This is in response to Terry Hall's question in ... Issue #7: "...do you know of an HDOS or CP/M program for making intermediate-sized type on an Epson...". It turns out Terry is within a few miles from me, but I thought others may be interested in my reply. For quite a few years I've been using a CP/M package sold by Skycastle Computer Products which allows you to print text-files, signs, what-have-you, in many many different fonts on an Epson printer (depending on your model, you can even download them). You can mix multiple fonts on the same line and you can logically AND/OR various backgrounds with the fonts. You can create a "frame" around your text, read in text from other files, create your own fonts (if you don't like the wide selection that they offer) and many other things. For more information, their address is P.O. Box 1412, San Carlos, California 94070.

[From Ralph Money, Omaha, NE, via OMAHUG's ONED BBS] Regarding Terry Hall [in the last issue] trying to run his setup with a -47 board in the right-hand (port 174Q) slot: I assume he jumpered SW501 so the monitor knows there's a -47 there, and set position 4 to whichever port he wants as the primary boot, and jumpered the -47 card to port 174Q. Oh, and installed the 4700 ohm resistor between pins 1 and 12 of P512 [if not using this port].

[From Charley Bejin, Cheyenne, WY] Reference the S.O.S. from G.R. Stradley in Issue No. 7 about using Turbo Pascal and SuperCalc2 on his H89. I use Turbo just a little but use SuperCalc2 extensively. I sent him some information directly, but thought one little item might be shared.

He mentioned operating at 4MHz. I have found that some 4MHz speed vendors suggest speeding up the terminal to 19,200 baud to take full advantage of the faster CPU speed. Some software, including SuperCalc2, will not run at 4MHz on my system using the Anaport speed modification and the terminal set to 19,200...

As you stated in your response to Mr. Stradley, Turbo has to be installed, and the CPU speed declared. My SuperCalc2 Version 1.00 has to be installed for my terminal, but there appears to be no provision to set the CPU speed. [I bought a close-out version of SuperCalc ver. 1.05 from Magnolia Microsystems two months. I discovered that it will run reliably at 4 MHz if you simply include the program that switches speed as the cold-boot command line in CONFIGUR. Because I'm running the TLB at 3 MHz and only 9,600 baud, I haven't checked what happens at a higher baud rate. But SC also apparently requires the customized version of CP/M on the disk it comes on, so you have to be careful on that score. -Ed.]

[From Hank Lotz, Pittsburgh, PA] ...[O]n T.J. Skollingsberg's FIRST item in his letter on page 2 of #7: I do not know about the SK-203 buffer, but if you are hooking up an H14 to it, I can tell you one thing I learned when hooking an H14 to an Angel buffer. The H14's Pin 4 gives a LOW READY signal, while Pin 15 of that same printer gives a HIGH READY signal. (Pin 20 of the printer cannot be used for that kind of READY SIGNAL; it merely indicates POWER ON, in the H14).

HDOS 3.0 co-author

[From Richard Musgrave, MIGHTY/SMART, Box 11164, Kansas City, MO 64119] I am the co-author of HDOS 3.0. As we discussed on the phone [Memorial Day weekend], I am interested in answering questions and/or writing about HDOS for your newsletter. I received Rick Streeter's HDOS mods and visited with him by phone the other day. I am disturbed by his many references to bugs. In fact, there are only 3 bugs in HDOS. **One** which he found ... All the other "bugs" he mentions are his incorrect assumptions about our source code...

Know bugs:

HDOS30.SYS -- .DISMT SYSCALL jumps to .DMOUNT by mistake; patch 015322A from 307Q to 173Q. Use PATCH.ABS from 3.0 utility disk.
 SYSCMD.SYS - "IF ERROR" in BATCH mode won't work because the error condition turns batch mode off. This has been fixed in a new release of SYSCMD which I have been working on.
 PIP.ABS - "/SAFE" switch displays wrong file if concatenating. Rick Streeter gets the credit for finding this one. His fix seems to work with no side effects.

I have been working on new releases of SYSCMD [and] PIP. A lot of Rick Streeter's ideas are being included and a lot of mine. New utilities which I have written:

MEGAPIP - A DOS shell like MS-DOS VFILER, PCBOSS, SWEEP, etc. A full-screen tag/untag, point [and] shoot tool. H19 required.
 DS - A 3.0 directory sort (similar to Norton's for MS-DOS).
 DFD - List deleted files in your directory (only their dir. entries).
 TWOCOPY - A silly abuse of ONECOPY that uses 2 drives. It does copy files faster [and] with less noise from the drives. Its kind of like XCOPY for MS-DOS.

I am working on a quick reference guide for HDOS 3.0. So far it's about 50 pages long. I will soon write an "ASSIST" utility which will give complete syntax [and] use of all commands [and] switches, but first I have to find the time. [Thanks, indeed, for your input, Richard. I'm looking forward to further material from you. -Ed.]

=====

THE 8-BIT IOWAN

by Kirk L Thompson

Even though I suggested (way back in issue #6) that I expected to review Heath's SK-203 printer buffer here, I'm going to put off that topic a while longer. More important to you and me, in my judgment, is the news I picked up at this year's CHUGCON.

CHUGCON88. This annual affair was held in Crystal City, VA, on the last weekend of July and was, as you and I should expect these days, mostly devoted to 16-bit systems. However, there was enough 8-bit information floating about, plus even several formal and informal vendors, that a roundup of it is worthwhile.

IN THE EXHIBIT AREA. I went as a vendor myself, taking subscriptions to **Staunch** (I picked up nine more!), selling keyboard overlays (disappointing!) and 8-bit p.d. software (some), and talking with anyone who stopped by my closet-sized booth. The only real trouble I had during the extended weekend was when the airline misplaced my box of merchandise! (It finally appeared three hours after the convention opened.) And the joy of the weekend was talking with those of you who came to the conference.

In fact, my first contact was with Computer Consultants to Business (111 S. George St., Cumberland, MD 21502, (301) 759-1260). This is a family-operated concern selling hardware and software for various systems. They have miscellaneous '89 items for sale: soft-sector controllers, 4MHz speedups, software, floppy drives, and books. Write for their flyer.

I also finally made contact with Robert Todd, distribution manager of SIG/M, custodian of that massive CP/M public domain library. (By coincidence, his GraphNet Systems booth was right across the aisle from my "closet.") It seems SIG/M is moving its mailing address; last winter I discovered its old Islin, NJ, box was closed. But I arranged with Bob to send him a postcard and he would mail it back with the new Reading, PA, box number. I hope to have that for you in the next issue.

More productive in that vein was my discussion with "Doc" Savage. He handles the SIG/M library for the "greater" St. Louis area (southern Illinois and eastern Missouri). He told me he distributes all SIG/M volumes in any 5" soft-sector format (DS/DD preferred) for \$2 each if formatted disks are provided, \$3 if no disks are supplied. Contact him at:

Robert G. "Doc" Savage / 112 Lakeland Hills Drive / Belleville, IL 62221-1321 / (618) 234-8221

Pat Swayne, HUG's illustrious software genius, also stopped by and mentioned that the MAGBASE program for CP/M he describes in his letter earlier in this issue is now available. The part number is 885-1249(-37) and price is, as I recall, \$20; within a week of my return home I put a check in the mail for it myself. I also talked briefly with him about the release of HDOS 2.0 source. Some of you have written or called HUG and been told they know nothing about it. Don't worry, Pat knows and if he does, you can bank on it!

One of the vendors I didn't expect there, but who made it nonetheless, was Borland International, represented by its Fairfax, VA, office. So I approached them about the status of the last version of Turbo Pascal for CP/M. I was told it was no longer in production. I had last seen it listed in Central Computer Products' 1988 CP/M catalog (the address is on the last page of issue #6). But if Central no longer has it and you are in need, call Borland directly at 1-800-543-7543. (This number is from a recent issue of Borland's **Turbo Technix**.) You may be shuffled around a bit because they will have to search through widely scattered warehouses, but persevere. The only compromise you may have to make is the media it comes on; standard H/Z formats may no longer be available, but there are people out here who can do media conversions for you, myself included.

I also talked with subscriber John Fitzpatrick about his "META-INPUT CONTROLLER (MIC)" device driver package for HDOS 2.0. He even left me an evaluation copy. This is a very versatile driver which serves as a friendly batch processor, like those built into HDOS 3.0 or MSDOS. The package can also run on an H-8 with the terminal operating off the H-8-5 card. It occupies 3K of memory. While there are other batch processors out there for HDOS, this is the first I've seen which accepts user-keyed input and permits conditional branching

and looping. It can be set up either for PROLOGUE.SYS or independent use.

The command files are prepared with an editor and can be used for simple command-driven operations (as you would with CP/M's SUBMIT) or preparing custom, menu-driven front-ends. Like SUBMIT, it permits use of command-line arguments in the command file. John includes examples for mounting and dismounting drives, running the assembler, swapping disks to list directories on the printer, command files for moving MIC files to other disks on both single- and multiple-drive systems, chaining Software Toolworks' PIE and TEXT together as a menu-driven word processor, and calling a Benton Harbor BASIC program while performing an automatic CTRL 4,1 to load the system overlays.

I fiddled further and discovered, as I expected, that the program to switch speeds on Kres Engineering's speed mod had to be last in the PROLOGUE.SYS command file (it apparently flushes the typeahead buffer). I also tried MIC with HDOS 3.0 and had the system lock up, so I can't recommend it there! In fact, the only (small) quibble I have with the package is that selection from custom menus is case-sensitive. You should use lower-case alphabetic characters or numbers for menu selection.

In any case, I recommend it for HDOS 2.0. The price is \$29.95 plus shipping and you can order it from:

John Fitzpatrick / Meridian Technical Assoc. Inc. /
2 Southboro Lane / Glen Rock, NJ 07452

A vendor who actually had a booth just down the aisle from mine was PAIR AND A SPARE, a surplus parts dealer. While there, proprietor (and new subscriber) Bruce Gewirtz bought Floppy Disk Services' entire stock of "TWO-ET" brackets. These, with the addition of some nuts and bolts, a power "Y" adapter, and appropriate data cable, simplify mounting two half-height drives inside the '89's cabinet. I hope to have more info on Bruce and his business next issue.

But as the second and final day of the conference wound to a close, CHUG and I swapped some software. I picked up Robbins' EDIT19, certainly the prime prize in CHUG's HDOS catalog. OTHER MATTERS. There were some noteworthy no-shows this year. Kres Engineering was certainly one. I also missed the presence of Lenny Geisler, editor of the friendly competition (SEBHC Journal). But most interesting was the absence of Barry Watzman marketing software for the '100. Indeed, I'm now hearing remarks from '100 owners about how lonely they are beginning to feel! MSDOS compatibility is apparently delaying what we have endured for some five years. But I mentioned to several individuals that the emergence of a '100-oriented newsletter in the next two years wouldn't surprise me at all.

Between the close of the exhibit area and the social hour before the banquet, I discussed CHUG and >CHUG with the latter's editor, Darrell Dean. Because of the high cost of putting out a monthly newsletter, CHUG anticipates surveying other computer groups to combine financial resources to publish and distribute a national user group newsletter with local inserts. If CHUG and OMAHUG are typical, a newsletter takes the largest chunk out of a local group's budget. But that kind of publishing scheme won't affect Staunch, of course. Most of what you and I see in local groups' rags are devoted to PC-compatibles. And that's not what you and I want to read about!

Finally, I actually made it to this year's banquet. You may recall Len Geisler's complaint (in his Nov. issue) about the "rubber chicken" at last year's banquet. (I missed that one because my plane luckily left the middle of Sunday afternoon.) Even though this year's hotel was different, the meal hadn't improved much. For one thing, adults at my

table were given children's chicken servings. For another (you guessed it!), the children got the adult plates. And to cap it off, once some of the kids' bounty was more equitably distributed, we discovered that bounty was mostly breeding! Maybe I should have ordered the steak.

But as is traditional at these occasions, the meal was stoically endured in anticipation of the speakers and what followed. There were two of the former this year, Alan Ashton, President of WordPerfect Corp., discussing his background and how he conceived his product, and a humorous history of Heath/Zenith computer systems and CHUG given by Mike Cogswell, now on the software staff at ZDS.

And after the speakers came, of course, the banquet door prize drawing, with a Z-386 as carrot. Perhaps fortunately, I did not take that home! Instead, I packed Andrew Tobia's: **Managing Your Money** in my suitcase early Sunday morning. Regrettably, this software is for PS/2's and on 3-1/2" disk to boot! I threatened to disassemble it and move it over. However, my "threat" was valueless and everyone grinned. (It will go, again as a door prize, at this month's OMAHUG meeting.)

But as I mentioned above, although the conference wasn't a financial success for me, it was still worth the trip. Talking with you, receiving your encouragement (and the odd gripe), and especially expanding circulation made the whole affair very enjoyable. And wouldn't you know, the airline did not lose anything on my return leg! But I knew I was back to reality when I stepped out into the blistering Iowa heat to find my car in the lot.

In the Queue. Next time, I'll depart a bit from my usual format. I've asked reader Terry Hall to write about his experiences with his harddisk system. Although prices for the hardware are falling, this is one route to increased productivity I haven't taken yet.

=====

USING HDOS 3.0

by Dr. Rick Streeter, D.O.S
4404 Willmary Ct., Kingsport, TN 37664

For the better part of the 1980s we had been hearing rumors: "HDOS3.0 is coming". It was not only to be an upgrade for existing hardware but also it was to give the H-100 something more to do with one of its dual processors in addition to CP/M. HDOS 3.0, though was quite a while a-comin'. Sometimes it seemed to be "vaporware". WELL, IT'S HERE!! And, the wait was worth it. For about the past two years Bill Parrott has been distributing the basic system (which covers seven H-17 disks). I sent him \$25.00 for the system and waited and waited. I thought my order was lost, but eventually it came. At that time it was rumored that HUG would only distribute the source on an MSDOS disk because of size. This would have been a real problem since HDOS 3.0 was useful mainly to an H-8 or H-89 user (us STAUNCH'ers); what would we be doing with an MSDOS machine! After all, wasn't that why we wanted HDOS. (In all fairness, some of us do have a TMSI H-1000 board, but what the h....) Bill Parrott came to the rescue and offered the source code for \$50.00. I ordered it and after waiting some more (six months) it came.

About HDOS 3.0: All of HDOS has been updated: all device drivers, system programs, support programs, BASIC, etc. Many new programs are provided to help document and demonstrate HDOS 3.0. There are seven screens of SYSHELP.DOC documenting new commands, and four screens describing PIP's about fifty (yes 50!) switches from "CHECK" to "VERSION". The directory structure has changed somewhat, and each file has four new file flag attributes: "A", "B", "D", and "U". "A"rchive for a yet-to-be-released archive utility, "B"ad for a verify utility also to be

HDOS MEMORY MAPS :
(Both presume memory expanded with WH-88-16 board)
Version 3.0
(Based on hardcopy provided by Rick Streeter)

1.000.000 (10000h) S.HIMEM+1	Monitor ROM & RAM Not accessible (8K)
377.377A (0FFFFh) S.HIMEM	DRIVERS, GRT TABLES, BUFFERS (Device drivers here are LOCKed)
357.377A (0EFFFh) S.RFWA	
S.RFWA-1	LOADed (but not LOCKed) DRIVERS (base of this is variable, if no DVDs exist then S.RFWA = S.SYSM)
357.377A (0EFFFh) S.SYSM	
S.SYSM-1	USER
042.200A (2280h) USERFWA	PROGRAM
041.177A (227Fh)	MEMORY
040.100A (2040h)	SYSTEM STACK (0.5 K)
040.077A (203Fh)	HDOS 3.0 work cells (formerly used by sys- tem MONITOR now main- tained by HDOS) (64 bytes)
040.000A (2000h)	
033.316A (1FFFFh)	BUFFERS AND WORK AREAS (1K)
033.316A (18CEh)	
033.315A (18CDh)	H-17 ROM (subroutines only) (1K)
030.000A (1800h)	
027.377A (17FFh)	HDOS30.SYS system Image (6K)
000.200A (0080h)	
000.177A (007Fh)	BASE PAGE (data, inter- rupt addresses, buffer pointers, etc.) (128 bytes)
000.000A (0000h)	

Version 2.0
(Based on HDOS 2.0 Manual & H-89A Operations Manual)

1.000.000 (10000h) S.HIMEM+1	1/2 of expansion memory not normally accessible (8K)
377.377A (0FFFFh) S.HIMEM	Resident HDOS system
337.377A (0DFFFh) SYSFWA	
SYSFWA-1	USER
042.200A (2280h) USERFWA	PROGRAM
042.177A (227Fh)	MEMORY
042.150A (2168h)	SYSTEM STACK (280 bytes)
041.147A (2167h)	SYSTEM RAM (360 bytes)
040.000A (2000h)	
037.377A (1FFFFh)	HDOS ROM (2K)
030.000A (1800h)	
027.377A (17FFh)	HDOS RAM (1K)
024.000A (1400h)	
023.377A (13FFh)	Expansion area for MTR- 90 and optional RAM chips in U522 & U524 (3K)
010.000A (0800h)	
007.377A (07FFh)	MTR-88, -89, & -90 (2K)
000.000A (0000h)	

VENDOR UPDATE

[Tell them you read it here!]

ZCPR. [From Richard A. Stanley, Wheeler Associates, Ltd., Box 9512, Alexandria, VA 22304] [I would first like] to make you and your readers aware of still another CP/M vendor: us! Enclosed is our descriptive sheet on ZCPR-IN, an auto-install (almost) approach to putting ZCPR3 on an H8 or H89/90. At the moment, the DR MACRO-80 assembler is required; however, if there is enough interest among your readers, we can modify the code so it can be assembled using a public domain assembler (which we would then include in the package).

Secondly, kudos on a good newsletter. Keep up the good work.

Lindley Systems. [From William Lindley] **LINDLEY SYSTEMS** has moved. Our new address is 4257 Berwick Place, Woodbridge, VA 22192, (703) 590-8890. Our printer drivers have been popular with HDOS and CP/M users, but we have not advertised them for several years. We would like to get these drivers out to as many of you as possible, so we are putting "clearance" prices on these items. For complete details, call or write for catalog.

Ultimate Driver. Control all the features of your printer. Use the tilde (~) character to insert control and escape codes in your files.

UPC Driver. Print H-19 graphics on your printer (or design your own characters). Will print double-height and/or double or triple width characters, also "reverse" white on black.

Specify printer (Epson and compatibles, NEC-8023, C.Itoh 8510, MPI, or Okidata), HDOS or Heath CP/M, and Hard or Soft sector disk. \$25 gets you two drivers (i.e., HDOS and CP/M versions of the Ultimate, or both drivers for HDOS), and \$35 gets you all four for your printer.

[Thanks for the information, Bill. And I enjoyed talking with you at CHUGCON. Readers may be interested in knowing that I prepared the keyboard-overlay-side of last issue's green insert with your UPC driver, using the default H-19 character set. I've also prepared a Greek set for printout on my C.Itoh 8510. -Ed.]

Another Micronics update. [From Darrell C. Pelan] Micronics Technology has moved to Montgomery! Our new address is: Suite 159, 54 Dalraida Rd., Montgomery, AL 36109. The phone number is 205-244-1597. Our phone hours are still 6-8 PM CST M-F and 9-12 Sat. The BBS is still up 24 hours, 300/1200/2400 at 205-244-0192. We continue our upgrades for the H-89. The 2/4 MHz **Speed Mod** is \$34.95 and the 20 meg hard disk, WIN89, is now **ONLY \$475 (\$450 cash)**. The WIN89 supports Heath, Magnolia, and CDR systems. HDOS support will be available soon. We have a very interested person working on the HDOS software (He has the WIN89 system and only runs HDOS).

Of course we still offer our software, **Perfect Printer, MT Accountant, and Paycheck** for the H-8 and H-89. We have a limited number of H-37 cards, original Heathkit equipment, with MTR-90 ROMs for \$150. Shipping is: **Speed Mods \$2, H-37 \$4, and WIN89 \$10.** Call or write for our free catalog....

Xpert Software. [8865 Polland Ave., San Diego, CA 92123, (619) 268-0112] You may recall that I mentioned this supplier last time. A catalog arrived several weeks after that issue went out to you. Products include a key remapper (which could be set up to perform multikey macros), a sideways or banner printer (requires Epson, C. Itoh, or Okidata printer), a dynamic debugger, and 68 public domain disks (editors, utilities, games, dBASE II utilities, communications, languages, etc.). The main caveat with this company is that its software is **not** available on hard-sector media, **only** soft-.

LONG LIVE THE H8/H89

Maybe you're the proud owner of a classic H8 and H19 terminal, lovingly built and maintained. Or perhaps you just purchased a used Zenith Z90. In any case, you already know what a great computer you've got.

I'm an electrical engineer, and fell in love with the H89 years ago. But like most engineers, I'm always looking for improvements. These products are the result. They're all compatible with any H8+H19 or H89 system. I use them myself, and personally guarantee your complete satisfaction or your money back.

✓ CPU BOARD ACCESSORIES

- Z89-37B SOFT-SECTOR CONTROLLER, assembled and tested\$195.00
 Exact replacement for Heath, but takes 1/2 the power. CPU board must have MTR-90, 444-61, 444-83, or equivalent (LP001).
 Z89-37U SOFT-SECTOR CONTROLLER with LP001 and all off-board parts and cables ..\$245.00
 Includes everything needed to upgrade older H8/H89s.
 Z89-37B SOFT-SECTOR CONTROLLER bare board, PALs, and manual\$50.00
 LP001 LOW-POWER KIT supplies MTR-90, 444-61, 444-83 in low-power CMOS parts\$29.95
 Includes MTR-90, cuts power by 250mA. Specify if non-Heath ROMs at U516-U520
 LP002 MTR-90A Monitor ROM (Heath MTR-90 with 38400 baud support)\$10.00
 Use with Superset to boot at 38400 baud for 4x faster screen operation.

✓ TERMINAL LOGIC BOARD ACCESSORIES

- FF001 FLICKER-FREE kit, the one that started it all!\$29.95
 Eliminates screen flicker and cuts power by 250mA for cooler operation.
 FF002 SUPERSET, the ultimate screen upgrade\$69.95
 Stock H19, HUG/Watzman, Super19, and UltraROM all in one! Includes FF001, four fonts (H19, VT-100, GT-PROM, math/greek), 38400 baud, screen saver, on-screen clock/calendar, help menus, reverse screen, and more.
 FF003 SUPERFONT adds IBM-PC, H29, VT-100 special symbols, 160x100 graphics\$29.95
 4 more fonts plus special symbols, double-wide characters, super/subscripts
 FF004 SUPERCLOCK adds battery backup for clock and 2nd page screen RAM\$29.95
 Adds screen save/restore, user-defined menus, fast animation, and windows!
 FF005 SUPERKEY uses Caps-Lock key to select "typewriter shift" operation\$15.00
 Don't need SHIFT key in BASIC, CP/M commands. Specify Superset or stock H19

✓ SOFTWARE (specify disk format)

- WHM01 WRITE-HAND-MAN, a "sidekick" for CP/M\$49.95
 Hit BREAK for instant access to a calculator, phonebook, notepad, calendar, view files, disk directory. NEW Superset version can capture screens from interrupted program; print, save, or edit and return them to your program!
 WHM00 WHM upgrade to latest version\$10.00
 AP002 Source codes for all WHM applications\$29.95
 Modify, write your own applications, or convert existing programs into memory-resident utilities. Requires Microsoft MAC80 assembler.
 SS003 SUPERFONT Source code to create your own fonts\$10.00
 Return changes on CP/M disk with \$10 for a custom Superfont ROM.

I am proud to announce that Quikdata is now a distributor of TMSI's Z89-37 Soft Sector Disk Controller, Write-Hand-Man, and Superset products. Henry Fale of Quikdata is one of the earliest supporters of the H8/H89, and has an excellent reputation for fast delivery, good pricing, and great service. For MC/VISA customers and quick delivery, order these products from Quikdata Inc, 2618 Penn Circle, Sheboygan WI 53081, phone 414-452-4172. Or order from me directly at:

TMSI c/o Lee Hart, 28612 Middle Crossing Road, Dowagiac, MI 49047, 616-782-3980

SUMMER CLEARANCE SALE

The lowest prices ever on the best H/Z89 hard disk systems!
Quantities are Limited!

Hard Disk Systems

*Sale
Prices*
↓

System #3 - 20 Megabytes with External Cabinet (65ms)	\$895 \$695
System #6 - 40 Megabytes with External Cabinet (40ms)	\$1145 \$895

These are complete packages which include the Hard Disk Drive, Western Digital WD1002 Hard Disk Controller, Universal Parallel Interface Board for the H/Z89, 50 Watt Power Supply, Drive Cabinet, Z37 compatible floppy disk controller, two Centronics parallel printer ports, complete software support package for both CP/M and HDOS, Sigma boot ROM, all required cables, 1 year warranty, and the Hard Disk System Users Manual.

System #7 - The Build-it-Yourself Package with Cabinet	\$645 \$545
--	------------------------

This package includes all of the above except the hard disk drive. Just add a ST225 or compatible drive to complete this system. Some drive experience is recommended.

System #8 - The Build-it-Yourself Package without Cabinet	\$495 \$375
---	------------------------

This package includes all of the above except the hard disk drive, cabinet, and power supply. A hard disk drive, drive cabinet, 40 Watt power supply, and some drive experience is required to complete this system.

Disk System Accessories

Hard Disk System Users Manual (100 pages)	\$10
Half Height Double Sided 40 Track Floppy Disk Drive	\$125
Half Height Double Sided 80 Track Floppy Disk Drive	\$175 \$145
Half Height 20 Megabyte 5.25 Inch Hard Disk Drive (65ms)	\$295 \$245
Half Height 40 Megabyte 5.25 Inch Hard Disk Drive (40ms)	\$495 \$395
External Drive Cabinet with 50 Watt Power Supply	\$195 \$175
Centronics Parallel Printer Cable (for most printer types)	\$45

All software is supplied on 5.25 inch, single sided, 40 track media. Please specify either hard or soft sector format with your order.

Domestic Shipping Charges

Ground Shipping for Software and/or Manuals Only	\$2
Ground Shipping and Insurance for Systems #7 and #8	\$5
Air Shipping and Insurance for Complete Disk Systems	\$20
COD Charge (must be approved in advance)	\$3

Please double these rates for delivery to areas outside of the United States via Air Mail. Texas residents please add 8% sales tax. Please include either a check, money order, or Visa/Mastercard charge authorization with your order.

SigmaSoft and Systems

17000 Dallas Parkway #207, Dallas, TX 75248, (214) 380-6187

From STUDIO COMPUTERS

(Courtesy of Ray Massa)
(and the authors noted below)

INCLUDES HARDCOPY DOCUMENTATIONFor HDOS Only**COMPACTA**

(Written by William Moss)

Allows storing up to a disk of files as a single file to conserve storage. Also permits packing system files (HDOS.SYS, HDOSOVLO.SYS, etc.) for later recovery. Similar to CP/M's LU and NULU, but does not "squeeze" or "unsqueeze." Particularly effective on high-capacity drives, but would be still more storage-efficient if a file squeezer (such as CRUNCH.ABS, available from SMUGH) were used. Two drives recommended. Source code included.

DATABASE

General-purpose database system specifically for Benton Harbor BASIC. Uses pseudo-random-access for searching and reading data. Also automatically constructs an index of records. Records based on 14-line "pages" of information, with each record containing up to 9 "pages." Two drives and 32K memory required. Printer not required for basic use, but routines included for hardcopies of data and index.

FINANCIAL

One program in this package calculates amortization, investment rate of return, and cost of borrowing. Output device is selectable (TT: [default], AT:, LP:, DD:, or BP:). The second program draws bar graphs on the selected device. Data to be graphed must be manually keyed, but up to 100 items may be entered. These two programs also "chain" to each other. Requires MBASIC and 48K memory.

MAILIST

A mailing list maintenance package for either Benton Harbor BASIC or MBASIC or both. Includes sorting and printing labels. Runs on either one or two drives; uses "stand-alone" for the former. Requires 48K memory.

PARALLELER

(Written by Tom Jorgenson)

A device driver for controlling your system from a remote terminal or control a remote terminal from your system, whether hard-wired or by modem. Occupies only 2.5K when loaded. Mass-storage device independent. Supports Hayes-compatible modems. 15 SET options and PROLOGUE.SYS, password protection, and auto-start of a user-supplied program. Could be used as the core of a BBS. Includes source code. Requires standard HDOS 2.0 and 32K memory.

PROBE

General purpose database package using random access techniques. Data entry built around an entry screen set up like a mailing list with 12 fields. Max. record length is 256 characters. Utilities included builds a compatible test file, removes records flagged deleted, recovers records flagged deleted, converts data files created under MAILIST (above), and creates a backup file index. Requires 56K memory, two drives, and MBASIC.

QUIZMASTER

Program to create, update, and give individually constructed tutoring. Content of the quiz is determined by the user. Requires Benton Harbor BASIC and 32K memory or MBASIC and 48K.

TERMINAL

(Written by Larry Boufford)

Menu-driven communications package for calling remote BBS's. Features auto-logon, Hayes compatibility, CTRL-S/CTRL-Q (XON/XOFF) protocol, and 2,000 character buffer. User-configurable baud rate, duplex, interrupts, and port. Status of carrier detect, aux. I/O, printer on/off, and free space displayed on 25th line. Does **not** support XMODEM file transfers. Source code not originally available, but I expect to have it shortly.

T.N.T DRIVER/DEMO

(Written by Tom Jorgenson)

Device driver for Votrax TYPE 'N TALK. With a custom cable and some care, a printer could be run off the same port as the Votrax. Includes a demo program and SPELL-IT game. The last requires MBASIC. Includes source code.

Additional Materials for CP/M only.**HANK LOTZ'S UTILITIES 3**

(Written by Hank Lotz)

MCOLS, a utility that creates multiple-column listings from a single-column input file. User specifies number of columns (2 to 13), space between them, effective page width and length and horizontal pitch, and record to start at. Includes MBASIC interpreted and compiled versions with documentation.

Magic Wand/PeachText HELP

(Written by Kirk L Thompson)

Menu-driven online HELP set up as an INCLUDE file for **Magic Wand** or **PeachText**. Command reference for both EDIT and PRINT. Rather sluggish on floppy-based systems; hard-disk and/or RAMdrive recommended. Documentation includes article (in MW/PT format) appearing in REMark (Aug, '88) telling how to prepare menu-driven include files, such as HELP. Also available for MSDOS.

Placing an Order

Except for the Studio Computers releases, your cost depends on what you supply:

Formatted disk(s) and self-addressed, stamped return mailer	\$2.00 per disk
Formatted disk(s) without mailer	\$4.00 per disk
No disk(s) or mailer	\$6.00 per disk

Because of the included hardcopy documentation, the packages from Studio Computers cost \$6 each even if the files are packed on high-capacity media. Formats available are standard (SS/SD) hard-sector and 40-track (48 tpi) soft-sector, single- or double-sided, for both HDOS and CP/M. Please clearly indicate the format you are supplying or require. If you desire soft-sector, I will pack multiple items onto one disk. Only the two CP/M disks can be packed and distributed for the single-disk charge. I will not subdivide a disk. Send mailorders to:

Kirk L Thompson / **The Staunch 8/89'er** / #6 West Branch Mob Hom Vil / Rte 1 / West Branch, IA 52358

released sooner or later, and "Delete protect which disallows deleting a file but allows reads and updates. You assembly language people will be able to use an .OPENU or an .OPENR scall on a "Delete-protected file but .OPENW won't work. A delete is considered by HDOS as a write. The last flag, "U", is a user defined flag. It is unused by any of HDOS 3.0's support programs. It can be used to flag a file for special processing, a file of special information, or whatever. I use it as a COPY flag to tell PIP which files to copy. I use the "Archive flag to tell PIP (and me) that a file has been COPYed.

Flags can be set and cleared from within PIP or SYSCMD, and if the SYSOP switch at byte twelve of SYSCMD.SYS is non-0, work is allowed on files which have the "L" flag set. Some things still can't be done to a "Locked file but it's flags can be changed. A CATalog has five options: "/List" (the one we are most used to), "/Full" (which shows the disk label and lots more disk and file information), "/Brief" (filenames only) for any printer or screen width, "/Groups" (lists files by groups in Hexadecimal, Decimal, and Octal), and "/Minimum" (a one column list). Directory displays can be sorted on five fields: Name, Extension, Creation date, Access date, and Access count. Files to be copied or otherwise affected can also be sorted on these same fields. A file may also be selected by a specific file attribute flag, the number of times it has been accessed by HDOS, a date before or after a specified date, the current date, and those older than a certain number of days. "NO" can precede flags and count selections. Wildcarded selections can be negated (those files are selected which do not fill the wildcard). A "SAFE" switch is available to check for a file's existence before it is overwritten, a "KEEP" switch will keep the former file's flags when it is overwritten by a file of the same name, and a "FORCE" switch will allow work on a file regardless of its flags.

HDOS 3.0 itself is markedly different. It is origin 0 which in this case means that HDOS is located at 0, not in high memory such as old HDOS or CP/M. Program entry is still at USERFWA (42200A/2280H) or above. The 8k of RAM in an H8 or H-89, and not recognized by pre-3.0 HDOS, is now used. This gives about 4-5k of additional user memory. No overlays are used and the .LOADO SCALL which now does nothing is not necessary in any program but, if it exists, is ignored. The device table is dynamically allocated and built at BOOTup to hold any number of device drivers. HDOS is now almost completely hardware independent with TT: as a separate .DVD.

HDOS 3.0 MEMORY ALLOCATION: The base page of memory, 000.000A to 000.177A [see the memory maps inside this issue's insert], contains HDOS flags, interrupt addresses and pointers to the system buffers: search PATH, system PROMPT, BATCH, COMMAND LINE, system DEFAULT BLOCK, etc. HDOS30.SYS code follows immediately from 000.200A to 027.377A. The H-17 ROM subroutines follow that at 030.000A to 033.315A. (Note that this is **just** the H-17 ROM subroutines, **none** of the H-17 driver code is there.) The memory cells which were formerly used by the system monitor are maintained by HDOS 3.0 at 040.000A to 040.077A.

This is followed by the HDOS data area at locations 040.100A to 042.177A. The user RAM memory at 042.200A (USERFWA) follows and extends to the variable location S.SYSM (the base of loaded but not locked drivers: S.SYSM to S.RFWA). Next, from S.RFWA to S.HIMEM (377.377A) are the GRT tables, locked drivers, and additional HDOS buffers.

Comparing this to HDOS 2.0 you can see that there are a few and quite important differences. Since the monitor and H-17 driver are overlayed by RAM and HDOS 3.0's system image, HDOS 3.0 cannot be

run on a machine unable to remap its lower 8k of memory from ROM to RAM. When Heath released CP/M, machines had to be able to remap memory, so they changed the memory address decoder (in the H-89 at U517). This feature was not used by HDOS until 3.0. All newer H-89s produced by Heath had this new ROM (444-66) but a few machines, notably some of those running old versions of Magnolia CP/M may still have the old U517 ROM (444-42).

DEVICE DRIVERS: Device drivers which process interrupts are very different in HDOS 3.0 and their use may crash a 3.0 system. I am presently exploring their format and will be reporting in the future how to write them. Non-interrupting drivers are little changed with the exception that the driver header code now contains a flag to HDOS 3.0 allowing the driver preamble (SET code) to be most any length.

COMPATIBILITY: Most pre-3.0 programs will work under 3.0 but a few will not. The ones which won't are usually *.ABS programs which either 1) need to be reassembled using HDOS 3.0 *.ACM files, 2) use H-17 ROM disk routines (not ROM subroutines), 3) use MONITOR routines (the MONITOR is no longer there), or 4) use the HDOS type-ahead buffer which is accessed differently in HDOS 3.0.

If you have the source to a non-working program it is rather easy to find the offending code and remove or correct it. A valuable source for replacement code is the source listing for MONITOR 88/89/90 which most people can find lurking under the dust on their shelves. Sometimes just hardcoding a ROM subroutine into the source will cure a most disgustingly sick program. In the future I will present other cures.

DIRECTORY STRUCTURE: I said earlier that HDOS 3.0 has a different directory structure. You probably heard that a HDOS 3.0 disk cannot be read by a pre-3.0 system. This is not quite true. While the length of the directory entries has not changed, pre-3.0 filenames were 13 bytes long, the filename and extension plus a PROJECT and a VERSION byte following them. These were unused, but there nonetheless and were read when searching for a file. These bytes had to be null since they weren't used. HDOS 3.0 has 11-character names. The next two bytes have been redefined as DIR.CTH (creation time in hours) and DIR.CTM (creation time in minutes). For pre-3.0 HDOS to read a 3.0 disk the two bytes after the filename and extension have to be changed to 00. HDOS 3.0 will treat the time as 12:00 am and other versions will see what they want to see and find the needed file.

A DISK PRECAUTION: HDOS 3.0 continues to be very protective of its media. Apparently since the source for previous versions has been available for so long, many vendors have been more or less careless about sticking to Heath standards in both the design of disk hardware and software. I have seen disks written by some of these controllers trashed before my eyes both with HDOS 3.0 and previous versions. I can only say that since there has been a version change, care should be exercised when writing to non-3.0 media though reads are ok. I can only advise the old admonition: "USE WRITE PROTECT TABS" on valuable media.

BATCH FILES: BATCH files in HDOS 3.0 are implemented. A BATCH file is an ASCII file of commands with a ".BAT" extension. This is read by SYSCMD and is acted upon as though the commands were typed at the console. In itself this is not really much different from CP/M's SUBMIT. But the BATCH processor is in addition a command interpreter. That is, it is a mini language with input polling, testing, branching, and prompting operators. Very complex commands can be constructed with this inter-

preter which will, with its command substitution, respond quite differently depending on the conditions encountered. BATCH files will be a topic covered in great detail in a later installment.

THE CLOCK: HDOS 3.0 stamps its files with a time in the directory display. This is implemented as an HDOS TASK. A TASK is very much like a pseudo-device driver such as those used under pre-3.0 for a "quick and dirty" method of modifying the operating system. 3.0 TASKS apparently were to be more fully supported as can be seen by the TASKDEF.ACM file, but as they are, they are quite attractive. They are maintained and protected by 3.0 and most all of the pseudo-drivers which were used under pre-3.0 could be re-written as a TASK. TASKs are usually smaller and easier to write.

IN CONCLUSION: There is much more to HDOS 3.0 than the small amount of information here. I will expand on the many topics I have talked about in future columns and would appreciate any suggestions on what I can cover in greater detail. If I have made mistakes, let me know (gently) and any suggestions of future HDOS 3.0 topics will be appreciated. I hope I have at least whetted your interest in this new HDOS and hope that you will get to really like it as do I.

HDOS 3.0 is available from:

Mr. William G. Parrott III
7010 Caenen Avenue Basic HDOS 3.0 system: \$25.00
Shawnee, KS 66216 Source code: \$50.00
=====

Adventures with a Disassembler by Hank Lotz

When you have a .COM file, the ideal way to make desired changes (or to just see how it works) is to look at the source code (for example, an MBASIC listing). But what if you purchased the software from a secretive vendor who didn't give you his proprietary source listing? This was my situation recently. The .COM file I had was very large. (I'll call it BIG.COM throughout, to preserve its anonymity.) Despite its size I thought even an **assembly-language** listing (.ASM file) would be better than nothing. So I set out to disassemble BIG.COM with the utilities from HUG's disk #885-1212. First, their UNLOAD utility to operate on BIG.COM and give a .HEX file (BIG.HEX), and then DISASM to derive from that .HEX file an assembly-language source (BIG.ASM). It worked (finally), but I ran into something else along the way -- an education!

The First Snag. Fat .COM files beget fat .HEX files and alas, the BIG.HEX file generated by UNLOAD would not fit on the same disk. Well, that didn't seem like much of a problem. I could just route the output file to another disk, right? Ordinarily, yes, but UNLOAD's documentation didn't say how to get the .HEX file to go to a different drive from the one the input file was on. Playing with command line arguments didn't work. UNLOAD, **itself**, can be on a separate drive. But it soon became unmistakably clear that its **input** and **output** files **must be together** on a single drive.

Birth of a Patch. But all was not lost. An assembly listing, UNLOAD.ASM, was supplied on the HUG disk. Now, what should I do to the program? The ideal strategy would be to make the output drive user-selectable from the command line -- and with some extra assembly coding, that could be done. But I wanted a quick solution, and luckily, with very few additions and changes to the assembly source I was able to make it force the BIG.HEX file to drive C:. You'll see below, that I named the new version

UNLOADC because it writes **only** to C:. Not what you'd call **flexible**, but hey, I put BIG.COM in B:, and a disk with lots of room in C:, and was quite happy with the results. Later I saw a simpler way to accomplish the **same** modification **without** changing UNLOAD.ASM. Merely applying a 4-byte patch to UNLOAD.COM with DDT does it! Here's the patch to make UNLOAD put its output .HEX file to another drive. What **you** type is in **boldface**. On some lines you type nothing. But hit RETURN after each line:

```
B>DDT UNLOAD.COM
DDT VERS 2.2
NEXT PC
0400 0100
-S010B
010B 09 08
010C 11
010D 5C 5D
010E 00
010F 21
0110 80 81
0111 03 .
-S0380
0380 00 03 (03=Drive C:, Use 01 for A:, 02=B:)
0381 20 .
-C (CONTROL-C)
B>SAVE 3 UNLOADC.COM (Note different file name)
```

Notice by typing a different byte into address 0380 you can have the constant output drive be A: or B:, etc., instead.

Well, with this, I threw my BIG.HEX file onto C: and ran DISASM creating the BIG.ASM file (with success, ultimately, but I must tell you more about that too, later in this account). Fortunately, DISASM **does** let you put its output on another drive, so we don't have to mess around patching DISASM.COM. But never fear, we'll arrive at enough other problem areas to keep us busy!

A Surprise Detour. Naturally it's no use modifying a newly disassembled program unless we can also assemble and load it again afterwards. To do **this** we use the CP/M distribution software, ASM.COM and LOAD.COM. First, the ASM.COM program assembles the assembly-language source file back into a .HEX file. Then LOAD is run to make a .COM file out of the .HEX file. Like DISASM.COM, ASM.COM will direct its output to whatever disk we tell it. But, to my surprise, I discovered that CP/M's LOAD program has the same sad property as HUG's UNLOAD. In LOAD.COM's case, it will only write its output .COM file to the **same drive** as its input .HEX file! This would never do for my project, because of the hugeness of BIG.HEX.

Since I successfully patched UNLOAD.COM to get around that kind of deficiency, I was hopeful I could do the same with LOAD.COM. But this time I did **not** have the .ASM source. This meant going back to UNLOAD and DISASM to make an **.ASM listing for LOAD.COM**, and I'm here to tell you, **that trip** led to the next installment of my education!

A Nested Detour. I knew, **before** disassembling LOAD.COM, that I'd get neither descriptive labels nor illustrative comments in the resulting source to guide me. But I ran UNLOAD on LOAD.COM, creating LOAD.HEX. Then I ran DISASM on the LOAD.HEX and got my LOAD.ASM. I asked DISASM to give its crude ASCII "comments" so I could spot ASCII string data. Not a necessary move, but it helps you to later make a more compact, organized, readable listing, in that you can more easily arrange string data into DB blocks. But before fooling around too much with my newborn LOAD.ASM, I first wanted to be sure I could eventually get it **back** into a .COM file. No good working on it if it wasn't going to reassemble. I ran ASM on it to check this out. To relieve the reader's suspense, it **does** reassemble (and reload) perfectly. I did not find that out, though, until I

investigated the 14 errors ASM displayed on the screen!

The error symbol was "S" in all cases. And "S" was not among the symbols listed in the CP/M 2.2.03 manual. With trembling hope, I tried the 2.2.04 manual, and it **did** explain the "S". It stood for "syntax error". Puzzling over the assembly code, I could see no syntax error. But I noticed all the errors were on "LXI H,addr" statements. Their opcode is 21 Hex. Remember I had asked DISASM for "comments"? Well, it interpreted each 21 Hex as a (!) and wrote that over in the comments field after each LXI H,addr instruction. The CP/M manual says a (!) character, in a source, causes ASM.COM to insert a line feed. (That feature lets you put multiple statements on one line, separated by (!)'s.) Each of my 14 error lines had ASCII garbage after the (!) in the comment field. That was because DISASM also translated the address info (the "addr" part) into ASCII characters which of course were meaningless. Later, when ASM.COM saw a (!), it tried to convert the ensuing garbage into a new instruction. Hence the syntax errors! Lines without garbage after the (!)'s were okay. I removed the (!) symbols and ASM ran smoothly. These 14 mishaps would not have occurred if I had answered "No" to DISASM's "WANT ASCII COMMENTS?". On the other hand, I would not have learned as much and could not have passed it along. And don't get me wrong, DISASM's "comments" feature is still useful to display in the .ASM file whatever legitimate alphanumeric data is embedded in the object code of a .COM file. Especially so, now that this side trip has taught us the caution required.

Now that this nested problem is cleared up we can revert to the perils of our **previous** detour, and try to remedy LOAD.COM's refusal to write to whichever drive we tell it.

Taming LOAD.COM. Well, there I was at last with a viable assembly source for the beast. I found strange things in it, like jumps to BDOS instead of CALL's, unlabeled instructions after RET's so that there was no path to them, consecutive RET's, and other tidbits. Obviously some deadwood was left in after past revisions -- and I used some of those areas for my patch. (The jumps to BDOS were themselves accessed by CALL's so that explained how a return to the proper address was effected.)

As for modifying LOAD.COM, I was not successful on my first tries. This was a difficult project and I spent many hours on it. It is interesting that although it is a **patch** that I present here, it was worked out through modifications to the disassembled source listing. (The disassembled source might just as well have been obtained with DDT, but I couldn't get a copy of the source onto disk from DDT.) When I tried inserting new instructions into the source, the program would fail. The reason was it expanded the code, and the existing object file was too position-sensitive because it had absolute-addressed buffers at the end. Rather than go through and change buffer addresses, I kept it the same size and made byte-for-byte changes. For every 3-byte instruction I put in, for example, I deleted 3 bytes from a "deadwood" area. Happily, this makes the modification easy to put in the form of a **patch**!

My first success yielded a LOAD.COM that would force a file to drive C:. When you ran it you had no other choice (just as you don't in my above patch for UNLOAD.COM). But I ended up refining the LOAD.COM patch so you can now specify the output drive in the **command line**, and it will write to any drive specified. If you do not specify, it goes to your default (logged) drive.

Using Bill Derby's CMP (compare) utility, I compared the distribution LOAD.COM shipped with CP/M 2.2.03 with that of 2.2.04 and found not one byte of difference. Therefore, whichever CP/M version you run, you can make this patch.

Since the patched version writes to any drive, I

suggest naming it LOADX.COM. The X is a good mnemonic for the variability of the destination drive name, and is not among CP/M's legal drives, A through P.

To illustrate usage, here are some sample command lines that assume you are loading MYFILE.HEX:

```
B>LOADX MYFILE B: (creates MYFILE.COM on B:)
B>LOADX MYFILE C: (creates MYFILE.COM on C:)
B>LOADX MYFILE (defaults MYFILE.COM to B:)
B>LOADX MYFILE C:NEW (creates MYFILE.COM on C:)
```

In the latter example the filename NEW is ignored. A drive name, **with the colon**, is **all** that is required in the second argument, same as if you were running the ED.COM program.

In all of the above lines, both MYFILE.HEX and LOADX.COM were on drive B:. Now suppose MYFILE.HEX was on drive C:.

```
B>LOADX C:MYFILE C: (creates MYFILE.COM on C:)
B>LOADX C:MYFILE (defaults MYFILE.COM to B:)
```

In the latter example the original unpatched LOAD.COM, by contrast, would have created MYFILE.COM on C: (consistent with my initial description of the situation). On the other hand,

```
C>B:LOADX MYFILE (defaults MYFILE.COM to C:)
```

because this time our **prompt** (the default drive) is C>.

The patch follows. As always, **you** type the **boldface**, and end each line by hitting RETURN:

```
B>DDT LOAD.COM      ---> -S0399
DDT VERS 2.2        0399 32 C3
NEXT PC             039A 1A 12
0800 0100            039B 0C 04
-S0101              039C C9 .
0101 40 B0           -S0412
0102 02 03           0412 21 32
0103 20 .            0413 2A 1A
-S03B0              0414 0C 0C (same)
03B0 21 F5           0415 70 F5
03B1 20 3A           0416 2B 3E
03B2 0C 6C           0417 71 00
03B3 70 00           0418 2A 32
03B4 2B 32           0419 29 5C
03B5 71 17           041A 0C 00
03B6 2A 04           041B EB F1
03B7 1F F1           041C 0E C3
03B8 0C C3           041D 17 9C
03B9 EB 40           041E CD 03
03BA 0E 02           041F 66 .
03BB 11 .            ---- -"C (CONTROL-C)
                        B>SAVE 7 LOADX.COM
```

If anyone actually **prefers** a LOADC.COM which, as you can infer from its title, **does unconditionally force** the .COM output file to drive C:, you can have it by using **only** the substitutions in **column 2** of this patch! Omit column 1, except for invoking DDT. But you must also replace the 00 in **address 0417** with 03 (for drive C:). 02 would give a **LOADB.COM**, and 01 a **LOADA.COM**. (By the way, when column 1 is **not** skipped, it doesn't really matter **what** goes into 0417!)

Splitting Up Big Files. A while back I warned you I'd talk more about the generation of the BIG.ASM file. The original BIG.COM file was about 30K. From this, UNLOAD (my "UNLOADC") wrote a BIG.HEX file of 74K. When I ran DISASM against that BIG.HEX, I reserved a 90K disk for the output BIG.ASM file. (I only have hard-sectored drives.) DISASM ran for over 3 hours on my 2-MHz H-89 and stopped when the disk was full (90K). I found that 1052 lines of BIG.HEX were processed (out of a total of 1681). I was able to piece the job, and the second "half" of BIG.ASM

was generated in 8 minutes resulting in a "part-2" file of 61K. Obviously, processing time is not a linear function of .HEX-file size!

My success in "piecing" this job implies I learned a few things: The .HEX file can be edited since it consists of ASCII records. Chop out its first umpteen-hundred already-processed lines but do leave the last few lines of that first .HEX-file section in, to cause a small overlap between parts 1 and 2 of your output .ASM file so you can verify continuity.

Part 2 of the .HEX file should have, as its first line, a line that "begins" with an opcode rather than the middle of an instruction, or the disassembler will be out of sync. To determine this, compare the last few lines of your "part 1" .ASM file with the last few .HEX-file lines that were processed. You'll need a hex table of opcodes unless you have them memorized (and other jokes). I said "begins with an opcode" and that bears some explanation. Each line (or record) in the .HEX file has certain other info before the actual instruction bytes start. Each line starts with a colon. Following the colon, skip the next 8 digits. The actual CPU instructions begin on the next digit. So, I repeat, be sure you are on track at that point with a 2-digit hex opcode and **not** a **continuation** of a 2- or 3-byte instruction from the previous record. And in checking that, beware: **the last 2 digits of every .HEX-file record comprise a checksum**, so **do not** consider the last 2 digits of any line as part of an instruction. If an argument of an opcode was continued from a previous record, it will have left off just **before** those last 2 checksum digits. (I myself would like to know more about the formation of this checksum, so you'll see I've submitted this question to the Q minus A column.)

So my original project, creating BIG.ASM, is complete except for the ponderous task of poring over and analyzing the voluminous source output. We're talking here in the order of magnitude of 10,000 lines of assembly-language source! But a handful of lines at the end of part 1 of BIG.ASM were indeed repeated at the start of BIG.ASM part 2, so I know I didn't lose anything. That's to emphasize my earlier advice to allow a bit of overlap in the two .HEX-file sections. For me, regardless of the further outcome of this adventure, it has already been worth it just for the education!

=====

Q minus A

(An open-forum "Q-only" column. Answers are solicited from readers!)

Q — On a cold bootup of an H-89 right after powerup, if you page thru memory viewing the bytes, you come upon a vast stretch of alternating groups of 00's and groups of FF HEX values, and later, various other values, even before you get to the CCP. How do these get there and what is the purpose of the configuration they are in? Would these patterns be identical if viewed in an H-8?

Q — The Heath literature says the keyboard encoder scans the keys on the keyboard looking for a closure. At what frequency is the scanning done?

Q — The last byte (2 hex chars) at the end of each record in a HEX file (output by the CP/M ASM.COM) forms a "checksum". How is the value of this checksum byte arrived at?

=====

MISCELLANY

Correspondence between PS: and Hank. [From Hank:] Dear Pat, ...Thank you for your response to

my STAUNCH 8/89'er write-up about ... bugs in HUG's HFM utility.... I installed [your fixes]... and they do indeed do the trick!

I [also] have a question [involving] the H-89 (or H-19) **function keys**.... Back in STAUNCH #2 I wrote up a method [page 7] by which a user can read the keys from his/her own program.... A copy of my article is enclosed....

If fast input is given, the routine can drop **one** character.... But once **one** char has been lost, **it does not happen again** that I have ever seen (until the program is rerun). The problem is minor but I am... curious about the cause. If you have any ideas, I'd like to hear them.... As [an] additional clue to the problem..., the lost char [can] appear **on the screen, after** the program ends (following a CP/M system prompt)....

[From Pat:] Dear Hank, The problem you described in CP/M, in which a character is absorbed and winds up beside the system prompt when your program exits, is due to a one-character type-ahead buffer maintained by the CP/M BDOS. This buffer is separate and apart from any type-ahead buffer maintained by your BIOS. If a character is typed when a character-reading routine, such as BDOS function 6, is not actively looking for one, that character is placed into the one-character buffer. It is then available only to BDOS functions 1 or 9, but not to function 6 or the BIOS input function. Therefore, if you type too fast for your function-key routine, a character could be grabbed by this one-character buffer, and it will appear to be lost to your program.

This one-character buffer can also cause a problem when you are using Control-S to stop and start a text display scrolling on the screen (as when you TYPE a large file). If you should accidentally type something besides Control-S while the display is scrolling, you may find that Control-S no longer works. That is because the BDOS only checks the one-character buffer for Control-S characters, and if it has already been filled with something else, it will never see a Control-S until the buffer has been emptied, which will not happen until your file has been completely displayed and the system prompt has reappeared.

The only real solution to this problem is to modify the BDOS to simply eliminate the one-character buffer and just rely on the BIOS buffer. I did just that in the emulated BDOS in my CP/Emulator program for MS-DOS, so that is one area in which it does not work EXACTLY like CP/M. Hopefully, the folks who have done CP/M enhancements such as Z-system also saw that "bug", and fixed it.

A pseudo-fix to the problem is to have your program call function 1 when it exits, after printing a prompt to "hit any key". If the buffer has a character in it, the program will exit without a key having to be hit. If the buffer is empty, the user will have to hit a key before the program will exit.

THE STAUNCH 8/89'er, created by Hank Lotz, is a quarterly newsletter on 8-bit H/Z computers. The editor is Kirk L. Thompson; #6 West Branch Mob Hom Vil; Route 1; West Branch, IA 52358. Subscriptions always start and end with the calendar year. Rate: \$8.00/year. (Overseas, add \$2.) Single copies: \$2. Make checks payable to "Kirk L. Thompson". **Staunch** pays authors for their articles; write for an author's guide. It also accepts commercial ads for a modest fee; contact the editor. Neither this newsletter nor its editor is responsible for damages or losses resulting from use of any information presented herein. Info from **THE STAUNCH 8/89'er** may be reprinted only if this publication's name and address is included. Credit should also be given to authors and other sources of said material, if known. CP/M is a registered trademark of Digital Research, Inc. REMark is a registered trademark of Heath/Zenith Users' Group. EOF