October 1978

CLOAD
MAGAZINE
BOX 1267
GOLETA, CA
93017

How to Speak on Half a Subject, or Piracy (cont...)

Last month I spoke a bit on the practice of copying other people's software and selling it, thus eliminating the tedious chore of writing the code. In my short-sightedness, I missed what is, to us, the worst part of the problem. The practice of selling it to us! last month, we landed in a briar patch on this point, not once, but twice. I'd like to take a paragraph or two to address this aspect of the problem, and (hopefully) clear the air for a while.

The first of the two incidences was my own personal fault, and here's my confession: An author who has contributed quite a bit of original material (good stuff, too) "threw in" a copy of Hamurabi in his last submission tape, with a note that the source was Dave Ahl's "101 Computer Games". I honestly don't remember seeing the note, though everyone else around here does. To make a short story shorter, I approved it, we bought it and published it. About three days after the September issue went out, Dave contacted us through his editor, John Craig, wondering what was going on (Dave Ahl is merely the publisher of the hottest personal computing magazine ever - Creative Computing). After some preliminary running around, we got together and worked the situation out to everyone's nominal satisfaction, though it took a fair amount of understanding from all parties to swing it. I should again stress that the person submitting it identified his source, and that the error was entirely my own. Those of you who would like to check out the source of this faux pas should purchase a copy of "102 Computer Games - Microcomputer Edition". It's a new version of the old classic "101". $8.25 ppd from Creative Computing, Box 789-M, rristown, NJ 07960, or try your local bookstore.

The second case was simpler, and more disturbing. The version of "Othello" that we published in our last issue was submitted to us as original work, and as we hadn't seen anything like it, we bought it, paid for it, and published it. The original author (Mr. Donald L. Dilley, of Federal Way, Washington) had sent a copy to Radio Shack, to Kilobaud Magazine, and to his son in southern California. This last copy was evidently sold with his son's computer system, thereby ending up in New York, from where it was submitted. We are assisting the author in tracking this down. We also have been informed that the name "Othello" when applied to this particular game, is a trade mark of CBS Toys. The game is centuries old, and the name has whiskers too, but when put together,they're private property.

So, folks, we have a new policy. All programs submitted to us must have the author's name and address inserted with "REM" statements in the first few lines of code. This should not bother the vast majority of authors, as they have a great deal of pride in their work. It should, however, discourage those who feel that they have "borrowed" too much code to truthfully call it their own work. Editing a program to run on the TRS-80 is not considered writing the code. We still need good material, and we still pay hard, cold cash up front for it (in increasing amounts), but we don't want other people's work, no matter how good or how cheap.

Enough.

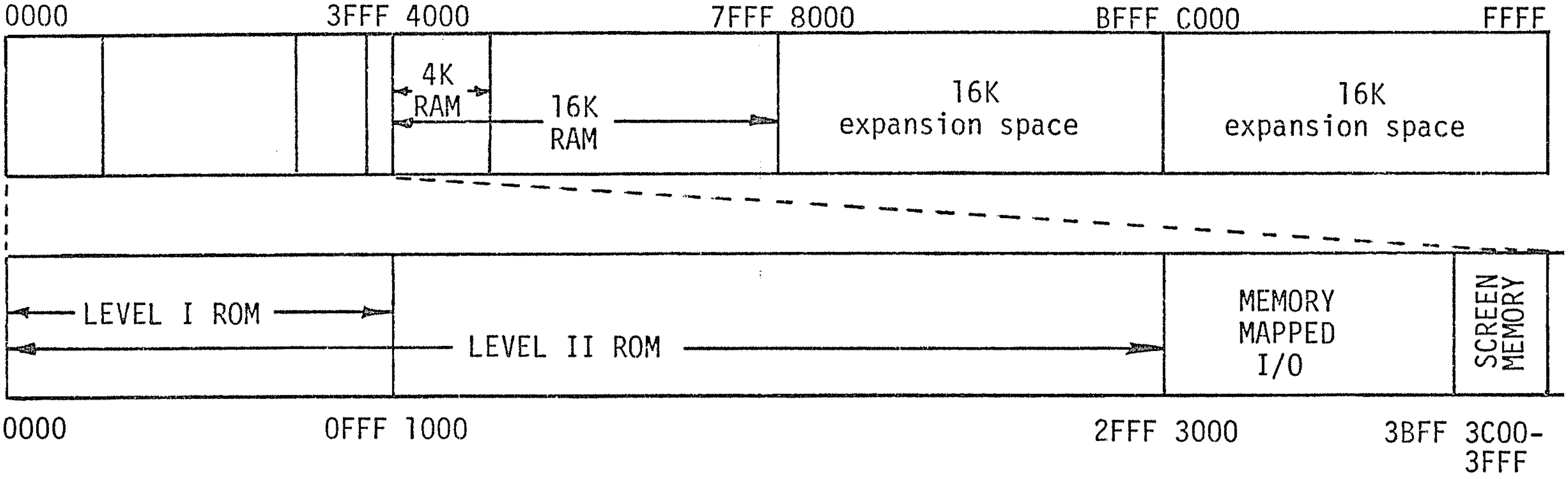
General announcements:

Those of you who have been resisting an urge to take your computer home at night, (or take it in to work in the morning), now have one less excuse. Ambico, Inc. of 101 Horton Ave, Lynbrook, NY 11563 has come out with a pair of cases for the TRS-80. Case "A" ($35) holds the TV monitor, and case "B" ($25) holds everything else. An introductory offer (good through 31 DEC 78) has both cases for $54.00 - give your computer a Christmas present.

Cload Magazine, in a valiant attempt at separating people from their money, is offering T-shirts with Clyde (that's him on the "Cload Information Sheet") emblazoned on the front. These are grade one, custom deluxe hi-cru shirts in bright yellow or light blue. If anyone wonders why you've been acting weird since your TRS-80 arrived, this shirt does all the explaining. $7.00 each, which includes shipping anywhere in the continental U.S. - CA residents add $.36 tax.

Donald Dilley informs us that the board in "Othello" can be redrawn by hitting "break" and then typing RUN 1000. He should know...

For this month's hardware column, I'd like to finish off memory and explain the program "MACHIN" on this month's tape.



The figure above is a "memory map". It shows in a visual manner the location of the various types of memory in the TRS-80. Let's take a hike through it, "Fantastic Voyage" style.

At address 0000 (Hex), we see the first byte of ROM. This is the first command in the machine, an F3 (Hex). It is the first command that the Z-80 chip executes after power-up (turning the computer on). What the F3 (Hex) does is tell the Z-80 to "disable interrupts", or ignore all signals coming in from the peripherals. Hiking through the ROM, we are looking at a rather long, complex sequence of these machine instructions. This sequence is a machine language program called the "BASIC interpreter". It does several things. One is to take input from the keyboard and put it into "user space" (which is also memory - we'll get there in a bit). It also puts it up on the screen so the programmer can see it. Another thing the BASIC interpreter does is CLOAD and CSAVE programs to the audio cassette deck. It does this by moving the contents of "user space" to the cassette or vice versa. The third major thing the BASIC interpreter does is to "RUN" programs. What it does then is to go to user space, grab the first line of the program, and "parse" it (interesting word - look it up in a dictionary). After picking it apart, it re-assembles it into a different sequence of machine instructions and executes it.

A fair estimate is that 90% of a program's run time is spent in the parse routine. This is one reason why BASIC has never caught on with professional programmers.

Those of you who are hiking through a Level 1 machine have by now reached the end of your ROM at address 0FFF (Hex). The memory space you're now hiking through has nothing in it at all, and looks like a bunch of FF (Hex) bytes. That's because if nothing pulls the data lines down to a 0, they float up to a 1, and 11111111 binary is FF (Hex).

Taking the bus to address 2FFF (Hex), we get off and observe the last byte of the Level II ROM. It's 00 (Hex) or "no operation", the command for the Z-80 to do absolutely nothing. I doubt if the Z-80 is ever allowed to execute this one. Walking into the next memory area, starting at 3000 (Hex), we see a few spiney outcroppings that look more like hardware than proper memory. This is the memory mapped I/O region and the things you see are latches and buffers for things like the keyboard, floppy disk controllers, line printers and the like. Before they're installed, of course, it just looks like empty memory - FF (Hex). At 3C00 (Hex) we arrive at our first read/write memory, a curious variety called screen memory. It has two unusual features. One is that bit 6 of each byte is always high. This is because there is no memory chip connected to it. Bits 0, 1, 2, 3, 4, 5 and 7 (the others) are all there. The other unusual thing about this memory is that it's visible. As we hike through, you will notice a large plate glass window to the right, looking out into the non-computer world. Some of you may see a rather large person staring back at you with a strange expression. This is the most complex component of the computer system, the programmer. They come in all sizes, shapes, colors, types of memory and degrees of sanity and sobriety. We won't be able to tour that section.

Our entry point, 3C00 H, is permanently attached to the upper left corner of the screen, and as we hike into higher address regions, we're traveling from left to right across the screen. After our 64th (Dec) byte (at 3C3F H) we start at the left side again, for another 64 (Dec) bytes. Notice that the bytes stored in these memory locations correspond to patterns. Those below 80 H in value correspond to letters of the alphabet, numbers and punctuation marks. You'll notice a lot of 20 H bytes in these storage locations. That's a blank, which is a "nothing" character to the programmer. There's no such thing as a "nothing" byte. The closest we come to it is a zero, or 00 H. It looks like an @. The bytes from 80 H up to FF H are geometric patterns which are used in the TRS-80 graphics mode.

At the end of the screen memory, location 3FFF H, we have hiked exactly one fourth of the way through memory space. Decimally speaking, we've passed through 16,384 bytes out of a total of 65,536.

We now pass into the 16,385th byte of storage. This is the first byte of true read/write memory, or the "RAM" that we spoke of last month. The first couple of hundred bytes are used as a sort of scratchpad area for the BASIC interpreter. Specifically, for Level I the scratchpad is from 4000 H to 41FF H, inclusive. In Level II, it's a bigger scratchpad - from 4000 H to 42E8 H. After the scratchpad area, there is a large area called "user space". This is where the BASIC program is stored. Those users who have 4K machines have 4096 (Dec) bytes of storage. Except for the scratchpad area, it is all user space. Those with 16K machines have 16,384 (Dec) bytes of storage. The end of this block is the halfway point of the total addressability of the Z-80 chip. If all the read/write memory possible is installed, there is 48K of it (49,152 bytes). It all looks the same as it does here at 4000 H, so the guided tour ends here. Please leave $2.75 by the power supply on the way out.

For those of you who are into machine language and/or assembler (using the T-Bug and Editor/Assembler packages, respectively), this last memory which starts at 4000 H is the area which holds the special software (TBUG or EDTASM) and the user-written software which is being developed.

As a demonstration, this month there is a machine language program called "MACHIN". Level I loading is via the normal "CLOAD" command. Level II loading of this, and all other machine language software, is via the "System" command. Type in "SYSTEM" and the TRS-80 will respond with a "?". Enter the name of the program ("MACHIN") and hit the enter key. This sequence of events is equivalent to typing "CLOAD" and hitting "enter". What the program does is put a 00 H in location 3C00 H, a 01 H in location 3C01 H, etc. It puts an FF H at location 3CFF H, and starts over again with a 00 H put in location 3D00 H. It works up through the screen memory this way, and spills a little into the "normal" RAM for reasons I'll discuss on a later date. This program will give you a feel for how fast the tape is loading programs, and what the Hex bytes look like on the screen. If you're really into binary patterns, you'll see why every other line is repeated.

See you next month!

Ralph McElroy

Ralph McElroy
Publisher

CLOAD INFO SHEET

Level I 4K programs - side I   Level II 16K programs - side II

All Level I 4K programs will run on a Level I 16K machine.

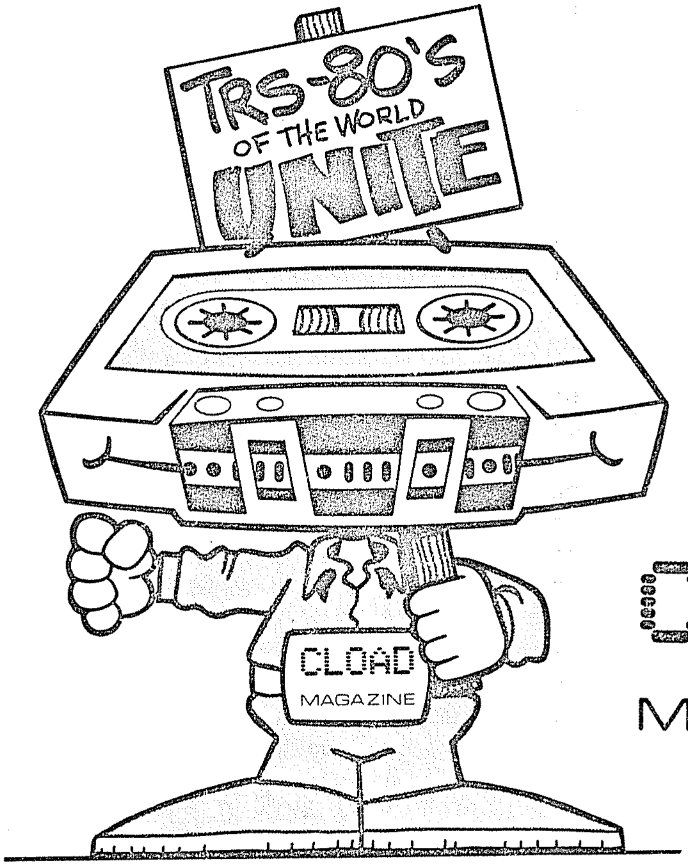
Some, but not all, Level II programs will run on a Level II 4K machine.

Subscriptions - $36.00/year (no tax).

Single issues $3.50 each, prepaid (please add 6% sales tax on orders from California). Level I issues available from March 1978. Tapes from June on have both Level I and Level II.

Playback level varies from machine to machine. Try around 7 or so and experiment. A small AM radio placed near the keyboard helps a great deal.

Numbers on the label are turns count. Rewind cassette completely, set counter to 000 and programs are easy to find.



CLOAD
MAGAZINE

Box 1267
Goleta, CA 93017
(805) 964-2761
MasterCharge/VISA
welcome