Olsen's Raid

An Update to the "Shiloh's Raid" Relative File Bug Fix

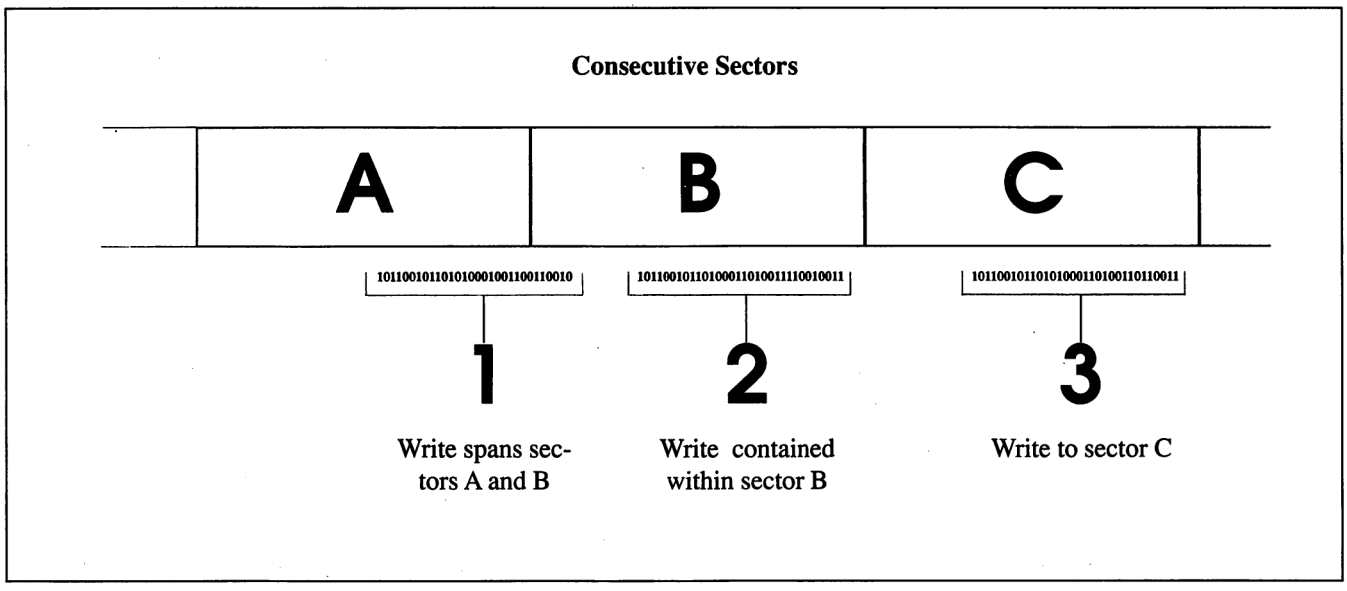
Volume 7, Issue 4 of *Transactor* contained an article by David Shiloh called "Shiloh's Raid" that claimed to eliminate the dreaded relative file bug: it showed under what circumstances the bug occurred, and took extra precautions when writing to the file under those conditions. Since that article appeared, we heard from Helen Olsen, who found a flaw in Shiloh's explanation of when the bug occurs. Helen sent us several programs to illustrate her point, and after hearing from her, and again from David Shiloh, we think it's time to clarify things a bit.

First, let's back-track a little. A generally-known bug in the 1541 causes problems under rare conditions when writing to a relative file. The fix (also generally known) is easy - just position the record pointer *twice* before writing a relative file record. The extra point (and some say a short delay as well) seems to eliminate the bug, so most experts advise to handle relative files in this way and eliminate the problem. David Shiloh took the extra step of finding under exactly what conditions the bug occurs, and applying the fix only under those conditions. The program he presented along with the article was supposed to prove that the fix works by checking for errors in a long random-write test both with and without the "Shiloh's Raid" routine in place.

Shiloh explained that the bug occurs under the following conditions - refer to the diagram below to illuminate the explanation: data is written to a relative file record, "spilling over" from one sector to another. This is write number 1 in the diagram, spanning sectors A and B. Write number 2 then takes place, to a record residing wholly within the next contiguous sector - sector B. According to Shiloh, the bug is now waiting to happen, and if a write (3) now occurs to the next sector (C), the data is instead erroneously written to sector A, potentially spilling into sector B as well.

Shiloh's solution was to detect when this condition was about to occur and apply the standard fix, to position the record pointer twice and pause before writing. The advantage to Shiloh's Raid is that the double-point need only be done on the rare occasions when the above circumstances occur, saving time for typical relative file access.

Enter Helen Olsen. Her main point was that the conditions that Shiloh sets for the bug to occur are too strict; only writes 1 and 3 (referring to the diagram again) need to take place to trigger the bug. She suggested that positioning an extra time (without the delay) after writing to a split record (1) is the best solution.



As it turns out, David Shiloh's explanation of when the bug occurs *is* too exacting, but the 'Shiloh's Raid' program works properly, that is, it senses trouble and does the extra pointer positioning even when only writes '1' and '3' occur. David Shiloh told us this, and shortly thereafter, Helen Olsen sent us a letter that concurred. Here is part of that letter:

> "To sum up my position on Shiloh's Raid: His description of the *cause* of the bug is wrong - his 1, 2, 3 sequence, with 2 setting up the bug is wrong. The bug happens with 1 followed by 3. *I* was wrong about his fix not working because I assumed that it was applied only in response to the 1, 2, 3 sequence. He may not realize it, but his subroutine repeats the positioning command often outside that sequence, including the 1, 3 sequence, which may be why he's unaware that *it* causes the bug, also... His fix is also applied when it is not necessary... In random use of a relative file, my fix, which repeats the position command after every write to a split record, will surely be used unnecessarily more often than Shiloh's fix, but since I don't know the cause of the bug (nor does he), I feel safer using it."

Helen presented programs along with her letters that proved her point, showing that only writes 1 and 3 were required to cause the bug, and showing that her fix worked.

So where does that leave us at the Transactor? Well, with a certain amount of egg on our face, to begin with, for not verifying that Shiloh's program was doing exactly what the text of the article said it was supposed to do. It may be even worse than that (for us), because the version of Shiloh's Raid that was printed was much improved cosmetically from the original - expanding from nineteen lines of tightly packed, unreadable code, to a page and a half of commented BASIC that had a chance of being understood. Perhaps something was lost in the translation that accounted for the application of the extra point-and-wait when it was unnecessary, i.e. outside of the "1, 2, 3" and "1, 3" conditions. In any case, we perhaps got carried away a tad in presenting the last word on the relative file bug, and we're glad that Helen brought us down to reality, though "glad" may not be the most appropriate word all around.

So, to summarize: The original Shiloh's Raid program was correct, in that it prevented the bug from occurring, and only applied the fix on a very small percentage of writes to the file. Shiloh's explanation of when the bug occurs doesn't cover all situations, so is only partially correct. Helen Olsen is right about the flaw in Shiloh's explanation, and her solution also seems to stop the bug from occurring. There is still no proof that there are no other ways in which the bug can occur, but it seems that you are safe if you use Shiloh's Raid, Olsen's fix, or if you just position twice before every write to a relative file.

Thus closes the file on Shiloh's Raid; as Helen Olsen ended her letter after announcing that it was to be her last on the subject, "Did I hear a heartfelt 'amen'?"

Announcing a unique new product for the C128

Jugg'ler - 128

By M. Garamszeghy

This program provides **read, write and formatting support for more than 130 types of MFM CP/M disks** on the C128 in CP/M mode with a 1570, 1571, or 1581 disk drive.

It is compatible with all current versions of C128 CP/M and all C128 hardware configurations including the 128-D. All normal CP/M file access commands can be used with the extra disk types.

Jugg'ler is available by mail order for $19.95 Canadian or $17.95 US from Transactor. Order from the card at the centre of this magazine.