Why did the old RAID database use a signed 16-bit integer for its record count? Why not unsigned, or a 32-bit integer?

devblogs.microsoft.com/oldnewthing/20210810-00

August 10, 2021

Raymond Chen

Some time ago, I noted that the RAID defect tracking system from the early 1990's had a limit of 32767 records. I noted that for large projects, like Windows 95, a new database had to be created when the bug count limit was reached. Why did the authors of RAID pick a signed 16-bit integer as their record counter? Why not an unsigned integer, which doubled the range, or a 32-bit integer, which took you to two billion?

You have to remember that all decisions are made in a particular context. At the time, bugs were tracked in a text file that everybody edited.[1] I don't know exactly, but let's say that there were a few hundred bugs in that file for Windows 1.0. The goal was to solve the problems and limitations of that system:

- Concurrent editing resulted in changes being lost.
- No reporting. "How many bugs are assigned to Chris?"
- No auditing. "Who edited the bug priority?"
- Low capacity.

A database with a front-end interface could solve these problems. And as for capacity, well, you're going to be running it on an IBM PC with 640KB of memory and an 8086 processor, so a 16-bit integer is a natural size for that architecture. And your target audience is a large project with *over a hundred issues*. A 16-bit integer will handle that with two orders of magnitude to spare!

Why not an `unsigned int`? What a ridiculous question. Windows 1.0 had a few hundred issues in its defect tracker. Reaching 32,767 records was unthinkable. You've already beaten the target by a factor of over 300 with the most natural data type for the language you're using. And if you use a back-of-the-envelope calculation that each record is about 1KB in size, your 20MB hard drive won't be able to hold more than 20,000 records anyway.

As I noted in the original article, the authors of the RAID bug tracking system had no intention of having their little hack become the *de facto* bug tracking system for the entire company for over a decade. They just wanted a way to track bugs in their day job in a manner that's a bit better than a shared text file. If they had known that they were designing the future of Microsoft internal bug tracking, they would have been too scared to write it!

[1] Sure, a text file seems unsophisticated, but it does the job, and it's a step up from using the inside of a Domino pizza box top (unnumbered page 20).

Raymond Chen

Follow