The strangest way of rounding down to the nearest quarter

devblogs.microsoft.com/oldnewthing/20050126-00

January 26, 2005

Raymond Chen

In a previous life, I wrote database software. A customer complained that one of their reports was taking an unacceptably long amount of time to generate, and I was asked to take a look at it even though it wasn't my account.

The report was a vacation-days report, listing the number of vacation days taken and available for each employee. Vacation days accrued at a fixed rate but were granted only in quarter-day increments. For example, if you earned 15 vacation days per year and the year was 32% complete, then you had accrued 32% × 15 = 4.8 vacation days, of which 4.75 were available to use.

The existing code to round the number of accrued days down to the nearest quarter-day went something like this:

```
* assume that at this point, ACCRUED is the number
* of accrued days.
PRIVATE S,F
* STR(ACCRUED,6,2) converts ACCRUED to a 6-character
* string: 3 integer digits, a decimal point, and two
* fractional digits.  Excess fractional digits are rounded.
STORE STR(ACCRUED,6,2) TO S
STORE RIGHT(S,2) TO F        && extract digits after decimal
IF F < "25"
 F = "00"                    && 00 to 24 becomes 00
ELSE
 IF F < "50"
  F = "25"                   && 25 to 49 becomes 25
 ELSE
  IF F < "75"
   F = "50"                  && 50 to 74 becomes 50
  ELSE
   F = "75"                  && 75 to 99 becomes 75
  ENDIF
 ENDIF
ENDIF
ROUNDED = VAL(LEFT(S,4) + F) && reconstruct value and convert
```

In other words, the code converted the number to a string, extracted the digits after the decimal point, did string comparisons to figure out which quartile the fraction resided in, then created a new string with the replacement fraction and converted that string back to a number. And all this in an interpreted language.

This code fragment was repeated each time rounding-down was needed because the language supported only 32 subroutines, and this procedure wasn't important enough to be worth kicking out one of the other existing subroutines.

I replaced this seventeen-line monstrosity with the one-line equivalent each time it occurred, and the report ran much faster.

(This is nowhere near the strangest way of implementing rounding. There are far worse examples.)

Exercise: What is the one-line equivalent?

Exercise: What is the double-rounding bug in the original code?

Raymond Chen

Follow