

When you have only 16KB of memory, you can't afford to waste any of it on fluffy stuff

 devblogs.microsoft.com/oldnewthing/20090128-00

January 28, 2009



Raymond Chen

The original IBM PC came with 16KB of memory.

That's not a lot of space in which to squeeze an operating system, especially since you had to leave enough memory for the user to actually *get work done*. A product of its time, the MS-DOS kernel is written entirely in assembly language, pretty much standard procedure for programs of the era. It also meant that the code takes all sorts of crazy shortcuts to shave a few bytes here, a few bytes there, in order to squeeze into as little memory as possible. For example, one very common trick was to have jump into the middle of an instruction, knowing that the second half of the instruction, when reinterpreted as the start of an instruction, performs the operation you wanted.

Anyway, this subject arose in response to my discussion of [why a corrupted program sometimes results in a "Program too big to fit in memory" error](#), which prompted the question from commenter 8 wondering [why the kernel didn't simply reject .COM files bigger than 64KB?](#)

Well, yeah, and that's what it did: By complaining that it was too big to fit into memory. There's no point adding a redundant test. (It appears that some people like to call these redundant tests *basic sanity checking*, but I consider sanity checking to be protecting against unreasonable values before they cause trouble. But in this case, they don't cause trouble—the error is detected and reported even without the so-called sanity check.)

Consider:

```
bool SomeFunction(...)
{
    ...
    if (x == 3) return false;
    if (x < 10) return false;
    ...
}
```

The first test is redundant, because if `x` is three, then even without the test, the function will still fail because `x` is also less than ten.

And when you're trying to squeeze your kernel into a few bytes as possible, you're certainly not going to waste your time coding up a redundant test.

Raymond Chen

Follow

