Detecting Existence of a Null Byte in a Multibyte Register

Borderite • 28 Dec 2025

Reading the assembly source code for *strcpy* in *glibc*, I found an interesting and efficient way to check if a multi-byte register contains a null byte. This document explains how it works.

Let X, Y, and Z be 64-bit registers. Suppose that we want to check if at least one among the eight bytes in X is zero. We set Y to a *magic number* and store the sum of X and Y in Z.

$$\begin{aligned} &Y \leftarrow \texttt{0xfefefefefefefeff}, \\ &Z \leftarrow X + Y. \end{aligned}$$

Suppose that each of the eight bytes in X is nonzero. In calculating the sum of X and Y, the zeroth byte apparently has a carry to the first byte. The second byte of Y plus the carry yields `0xff`, so that the first byte has a carry to the second. We can analogously verify each of the zeroth through sixth bytes has a carry to the text byte of it. For the same reason, the seventh byte sets the carry flag (CF).

On the other hand, if one byte among the eight bytes in X is zero, calculation of $X + Y$ does not yield a carry from the particular byte to the next, even if there is a carry from the previous byte. This means that X has no null byte if and only if every byte has a carry to the next (or CF) in calculation of $X + Y$.

While it is easy to check if CF is set, how can we check if each of the first through seventh bytes gets a carry in calculation of $X + Y$? Note that the last significant bit of `0xfe` is zero. Thus, a byte has a carry from the previous one if and only if the least significant bit of the byte is flipped in the addition. We thus calculate the bitwise XOR of X and Z and store the result in Z:

$$Z \leftarrow X \wedge Z.$$

Also, note that `0xfe` masks in all bits in a byte but the least significant one. Let's take the bitwise OR of Z and Y and store the result in Z:

$$Z \leftarrow X|Z.$$

After this operation, the register X contains a null byte if and only if CF is set and all bits of Z are ones. An easy way to check the latter condition is to test if CF is set and then test if addition of one to Z yields zero.

Although the current discussion assumed X, Y, and Z are 64-bit registers, the technique described above naturally extends to multibyte registers of any size.