

The curse of the leading zero

 devblogs.microsoft.com/oldnewthing/20140116-00

January 16, 2014



Raymond Chen

Consider the following batch file which tries to decide whether we are in the first or second half of the calendar year. (Assume US-English locale settings.)

```
if %DATE:~4,2% LEQ 6 (  
    echo First half  
) else (  
    echo Second Half  
)
```

This works great, except that it reports that August and September are in the first half of the year. What the heck? Have the laws of mathematics broken down?

Or this JavaScript function that creates a table of known postal codes for Cambridge, Massachusetts.

```
var CambridgeMA = [ 02138, 02139, 02140, 02141, 02142, 02238 ];
```

But when you try to use the array, you discover that half of the numbers got corrupted!

```
alert(CambridgeMA.join(" "));
```

```
| 2138 2139 1120 1121 1122 2238
```

Are space aliens corrupting my data?

Here's a clue. If you try to calculate the next month in a batch file

```
set /a NEXTMONTH=%DATE:~4,2%+1
```

the script generates the following error in August and September:

```
Invalid number. Numeric constants are either decimal (17),  
hexadecimal (0x11), or octal (021).
```

The answer is that pesky leading zero. (August is month 08 and September is month 09.)

Remember octal? I don't.

The architectural design of the PDP-8 and other processors of the era made octal a convenient notation for representing values. (This octal-centricness can also be seen in the instruction set of the 8008 processor, which led to the 8080, which led to the 8086, which led to the x86, which led to the x64, and you can still see the octal encoding in the so-called ModR/M and SIB bytes.)

The B language permitted octal constants to be expressed by prefixing them with a zero. For example, `010` represented the decimal value 8. This syntax was carried forward into the C language, and from there to every C-inspired language or language designed by people who imprinted on C.

Nowadays, octal is very rarely used, and as a result, the ability to create an octal constant by inserting a leading zero is now a curse rather than a helpful feature.

Now I can tell a joke: There's a company that apparently has difficulty printing decimal values, because a value like "30.038" ends up being printed as "30. 38". I suspect that their reporting program has a function like this:

```
void PrintThousandths(unsigned n)
{
    printf("%d.%03d", n / 1000, n % 1000);
}
```

One of my colleagues imagined what the code review must have looked like:

```
printf("%d.%03d", n / 1000, n % 1000);
```

[reviewer] Change `%03d` to `%3d`, because `%03d` will print the result in octal.

Raymond Chen

Follow

