

Switching on a tuple: Sneaky trick

 devblogs.microsoft.com/oldnewthing/20150326-00

March 26, 2015



Raymond Chen

This is a sneaky trick, but it's sometimes a handy sneaky trick.

Suppose you have two values, and you want to switch on the tuple. In other words, you want a switch statement for something like this:

```
if (val1 == 1 && val2 == 0) {
    Thing_1_0();
} else if ((val1 == 1 && val2 == 1 ||
           (val1 == 1 && val2 == 2)) {
    Thing_1_12();
} else if (val1 == 2 && val2 == 0) {
    Thing_2_0();
} ... etc ...
```

You could try writing

```
switch (val1, val2) {
case 1, 0:
    Thing_1_0();
    break;
case 1, 1:
case 1, 2:
    Thing_1_12();
    break;
case 2, 0:
    Thing_2_0();
    break;
...
}
```

but that doesn't do what you think it does. (Because that comma is a comma operator.)

The sneaky trick is to pack the two values into a single value that you can then switch on.

```
switch (MAKELONG(val1, val2)) {
case MAKELONG(1, 0):
    Thing_1_0();
    break;
case MAKELONG(1, 1):
case MAKELONG(1, 2):
    Thing_1_12();
    break;
case MAKELONG(2, 0):
    Thing_2_0();
    break;
...
}
```

Note that there are dangers here beyond craziness. You have to make sure that your packing function is injective (i.e., that it does not assign the same packed value to two different inputs). If you use `MAKELONG` as your packing function, then the two values to be packed must fit into 16-bit integers.

[Raymond Chen](#)

Follow

