

Q & A

Q: What does the `NULL` macro represent? [p. 415]

A: `NULL` actually stands for 0. When we use 0 in a context where a pointer would be required, C compilers treat it as a null pointer instead of the integer 0. The `NULL` macro is provided merely to help avoid confusion. The assignment

```
p = 0;
```

could be assigning the value 0 to a numeric variable or assigning a null pointer to a pointer variable; we can't easily tell which. In contrast, the assignment

```
p = NULL;
```

makes it clear that `p` is a pointer.

***Q:** In the header files that come with my compiler, `NULL` is defined as follows:

```
#define NULL (void *) 0
```

What's the advantage of casting 0 to `void *`?

A: This trick, which is allowed by the C standard, enables compilers to spot incorrect uses of the null pointer. For example, suppose that we try to assign `NULL` to an integer variable:

```
i = NULL;
```

If `NULL` is defined as 0, this assignment is perfectly legal. But if `NULL` is defined as `(void *) 0`, the compiler can warn us that we're assigning a pointer to an integer variable.

Defining `NULL` as `(void *) 0` has a second, more important, advantage. Suppose that we call a function with a variable-length argument list and pass `NULL` as one of the arguments. If `NULL` is defined as 0, the compiler will incorrectly pass a zero integer value. (In an ordinary function call, `NULL` works fine because the compiler knows from the function's prototype that it expects a pointer. When a function has a variable-length argument list, however, the compiler lacks this knowledge.) If `NULL` is defined as `(void *) 0`, the compiler will pass a null pointer.

To make matters even more confusing, some header files define `NULL` to be `0L` (the long version of 0). This definition, like the definition of `NULL` as 0, is a holdover from C's earlier years, when pointers and integers were compatible. For most purposes, though, it really doesn't matter how `NULL` is defined; just think of it as a name for the null pointer.

Q: Since 0 is used to represent the null pointer, I guess a null pointer is just an address with all zero bits, right?