

Dereference operator

From Wikipedia, the free encyclopedia

The **dereference operator** or **indirection operator**, denoted by "*" (i.e. an asterisk), is a unary operator found in C-like languages that include pointer variables. It operates on a pointer variable, and returns an l-value equivalent to the value at the pointer address. This is called "dereferencing" the pointer. For example, the C code

```
int x;  
int *p; // * is used in the declaration:  
        // p is a pointer to an integer, since (after dereferencing),  
        // *p is an integer  
  
x = 0;  
// now x == 0  
p = &x; // & takes the address of x  
// now p == &x, so *p == x  
*p = 1; // equivalent to x = 1, since *p == x  
// now *p == 1 and *p == x, so x == 1
```

assigned 1 to variable **x** by using the dereference operator and a pointer to the variable **x**.

Composition

The unary * operator, as defined in C and C++, can be used in compositions in cases of multiple indirection, where multiple acts of dereferencing are required. Pointers can of course reference other pointers, and in such cases, multiple applications of the dereference operator are needed. Similarly, the Java dot operator can be used in compositions forming quite sophisticated statements that require substantial dereferencing of pointers behind the scenes during evaluation.

A basic example of multiple pointer indirection is the argv argument to the main function in C (and C++), which is given in the prototype as char **argv. The name of the invoked program executable, as well as all command line arguments that followed, are stored as independent character strings. An array of pointers to char contains pointers to the first character of each of these strings, and this array of pointers is passed to the main function as the argv argument. The passed array itself "decays" to a pointer, thus argv is actually a pointer to a pointer to char, even though it stands for an array of pointers to char (similarly, the pointers in the array, while each formally pointing to a single char, actually point to what are strings of characters). The accompanying main argument, argc, provides the size of the array (i.e. the number of strings pointed to by the elements of the array), as the size of an (outmost) array is otherwise lost when it is passed to a function and converted to a pointer. Thus, argv is a pointer to the 0th element of an array of pointers to char, *argv, which in turn is a pointer to **argv, a character (precisely, the 0th character of the first argument string, which by convention is the name of the program).

Other syntax

In BCPL, an ancestor of C, the equivalent operator was represented using an exclamation mark.

In C, there is syntactic sugar for accessing members of a struct or union, given a pointer to such. Given a pointer `p` to a structure `s` so:

```
*p = s
```

the usual way to access a member `a` is as `s.a` which, given the pointer, is expressed as `(*p).a` or can instead be accessed by the shorthand:

```
p->a
```

This can be chained; for example, in a linked list, one may refer to `n->next->next` for the second following node (assuming that `n->next` is not null).

In Unix shell scripting and in utilities such as Makefiles, the dollar sign "\$" is the dereference operator, used to translate the name of a variable into its contents, and is notably absent when assigning to a variable.

In various languages, prefixes are used in identifiers, known as sigils. These are not unary operators – syntactically they are lexically part of the identifier, and have different semantics, such as indicating the data type of the identifier – but are syntactically similar to the dereference operator and can be confused with it. For example, in a shell script `$FOO` is the dereference operator `$` applied to the variable `FOO`, while in Perl `$foo` is a scalar variable called `foo`. In PHP, `FOO` is a constant (user defined or built-in), `$FOO` is a variable named `FOO` and `$$FOO` is a variable, whose name is stored in variable named `FOO`.

See also

- Segmentation fault

Retrieved from "https://en.wikipedia.org/w/index.php?title=Dereference_operator&oldid=683356380"

Categories: Operators (programming)

-
- This page was last modified on 29 September 2015, at 19:52.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.