Different sizeof results

Asked 16 years, 3 months ago Modified 7 years, 11 months ago Viewed 1k times

18

Why does `n` not equal to `8` in the following function?

```
void foo(char cvalue[8])
{
  int n = sizeof (cvalue);
}
```

But `n` *does* equal to `8` in this version of the function:

```
void bar()
{
  char cvalue[8];
  int n = sizeof (cvalue);
}
```

c++ c sizeof

Share
Improve this question
Follow

edited Jan 4, 2017 at 12:51
Kirill Kobelev
10.5k ● 6 ● 32 ● 52

asked Sep 8, 2008 at 2:31
jholl
2,084 ● 2 ● 18 ● 22

4 Answers

Sorted by: Highest score (default)

49

Because you can't pass entire arrays as function parameters in C. You're actually passing a pointer to it; the brackets are syntactic sugar. There are no guarantees the array you're pointing to has size 8, since you could pass this function any character pointer you want.

```
// These all do the same thing
void foo(char cvalue[8])
void foo(char cvalue[])
void foo(char *cvalue)
```

Share
Improve this answer
Follow

edited May 20, 2009 at 11:04

answered Sep 8, 2008 at 2:35
Nick Retallack
19.5k ● 19 ● 94 ● 115

15

C and C++ arrays are not first class objects; you cannot pass arrays to functions, they always decay to pointers.

You can, however, pass pointers and references to arrays. This prevents the array bounds from decaying. So this is legal:

```
template<typename T, size_t N>
void foo(const T(&arr)[N])
{
    int n = sizeof(arr);
}
```

Share Improve this answer Follow

answered Sep 8, 2008 at 13:03
DrPizza
18.3k ● 7 ● 42 ● 53

2 You deserve more upmods for your clever solution. – Nick Retallack May 20, 2009 at 11:03

1 Just need to note that the template example is C++, not C. C does not support reference types. – Peter Dec 21, 2015 at 12:16

1

In the first example, cvalue as passed parameter is in really just a pointer to a character array and when you take the `sizeof()` of it, you get the size of the pointer. In the second case, where you've declared it as a local variable, you get the size of the the entire array.

Share
Improve this answer
Follow

edited Dec 22, 2015 at 20:23
LogicStuff
19.6k ● 6 ● 56 ● 74

answered Sep 8, 2008 at 2:41
dagorym
5,795 ● 3 ● 26 ● 23

0

The size of the parameter on 32-bit systems will be 4 and on 64-bit systems compiled with -m64 will be 8. This is because arrays are passed as pointers in functions. The pointer is merely a memory address.

Share Improve this answer Follow

answered Sep 8, 2008 at 3:39
hoyhoy
6,351 ● 7 ● 40 ● 36