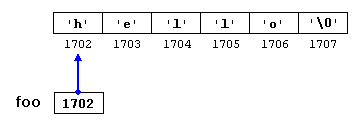
**Pointers and string literals.**

As pointed earlier, *string literals* are arrays containing null-terminated character sequences. In earlier sections, string literals have been used to be directly inserted into *cout*, to initialize strings and to initialize arrays of characters.

But they can also be accessed directly. String literals are arrays of the proper array type to contain all its characters plus the terminating null-character, with each of the elements being of type const char (as literals, they can never be modified). For example:

|  |  |  |
| --- | --- | --- |
|  | const char \* foo = "hello"; |  |

This declares an array with the literal representation for "*hello*", and then a pointer to its first element is assigned to *foo*. If we imagine that "*hello*" is stored at the memory locations that start at address 1702, we can represent the previous declaration as:



Note that here *foo* is a pointer and contains the value 1702, and not *'h'*, nor *"hello"*, although 1702 indeed is the address of both of these.

The pointer *foo* points to a sequence of characters. And because pointers and arrays behave essentially in the same way in expressions, *foo* can be used to access the characters in the same way arrays of null-terminated character sequences are. For example:

|  |  |  |
| --- | --- | --- |
| 1 2 | \*(foo+4)  foo[4] |  |

Both expressions have a value of 'o' (the fifth element of the array).