

深入解析C++中sizeof和strlen的奥秘：区别、应用与技巧全揭秘！

作者：微信公众号：【架构师老卢】

11-27 7:21

138



概述：`sizeof` 和 `strlen` 是 C++ 中用于处理字符串的两个不同的操作符，它们的作用和使用场景有很大的区别。

`sizeof` 和 `strlen` 是 C++ 中用于处理字符串的两个不同的操作符，它们的作用和使用场景有很大的区别。

sizeof 操作符：

区别：

- `sizeof` 是一个运算符，不是一个函数，用于获取一个类型或变量的字节大小。
- 对于数组，`sizeof` 返回整个数组的字节大小。
- 对于指针，`sizeof` 返回指针本身的字节大小，而不是指针指向的内存块的大小。

应用和示例：

```
1 #include <iostream>
2
3 int main() {
4     int integerVar;
5     double doubleVar;
6     char charArray[10];
7     char* charPointer;
8
9     std::cout << "Size of int: " << sizeof(integerVar) << " bytes" << std::endl;
10    std::cout << "Size of double: " << sizeof(doubleVar) << " bytes" << std::endl;
11    std::cout << "Size of char array: " << sizeof(charArray) << " bytes" << std::endl;
12    std::cout << "Size of char pointer: " << sizeof(charPointer) << " bytes" << std::endl;
13
14    return 0;
15 }
```

上述代码中，`sizeof` 被用于获取不同类型和变量的字节大小，输出结果将显示它们在内存中占用的空间大小。

strlen 函数：

区别：

- `strlen` 是一个函数，用于计算字符串的长度，即字符串中的字符数，不包括终止符 `\0`。
- 对于字符数组，`strlen` 返回数组中第一个字符到 `\0` 终止符的字符数。
- 对于指针，`strlen` 返回指针指向的字符串的长度，同样不包括终止符 `\0`。

应用和示例：

```
1 #include <iostream>
2 #include <cstring>
3
4 int main() {
5     const char* str = "Hello, World!";
6
7     std::cout << "Length of \"" << str << "\": " << strlen(str) << " characters" << std::endl;
8
9     char charArray[] = "C++ is fun!";
10    std::cout << "Length of \"" << charArray << "\": " << strlen(charArray) << " characters" << std::endl;
11
12    return 0;
13 }
```

上述代码中，`strlen` 用于获取字符串的长度，输出结果将显示每个字符串中的字符数。

注意事项：

1. **sizeof 和数组：** `sizeof` 返回整个数组的字节大小，而不是数组中的元素个数。如果需要获取数组中的元素个数，可以使用 `sizeof(array) / sizeof(array[0])` 的方式。
2. **strlen 和字符串终止符：** `strlen` 计算字符串的长度时，不包括字符串终止符 `\0`。因此，实际存储的字符数是 `strlen` 返回的值加上终止符。
3. **指针的 sizeof：** 对指针使用 `sizeof` 会返回指针本身的字节大小，而不是指向的数据的大小。要获取指针指向的数据大小，需要使用其他方式，如通过解引用指针后再使用 `sizeof`。

`sizeof` 主要用于获取数据类型或变量的字节大小，而 `strlen` 用于获取字符串的长度。在使用这两者时，需要注意其返回值的含义和用途，以避免出现错误。