

## 高效遍历：C++中分隔字符串单词的3种方法详解与实例

作者：微信公众号：【架构师老卢】

12-18 11:57

135



**概述：**在C++中，遍历由空格分隔的字符串的单词有多种方法，包括使用`std::istringstream`、手动遍历字符和正则表达式。其中，`std::istringstream`是简单高效的选择，通过流提取单词。手动遍历字符较为繁琐，正则表达式方法更灵活但可能有性能开销。根据实际需求选择方法，本文提供了清晰的实例源代码。

在C++中，遍历由空格分隔的单词组成的字符串有多种方法，其中包括使用C++标准库中的`std::istringstream`、手动遍历字符的方法以及使用正则表达式等。下面将分别介绍这些方法，并提供详细的实例源代码。

### 方法一：使用 `std::istringstream`

```
1 #include <iostream>
2 #include <sstream>
3 #include <string>
4
5 int main() {
6     std::string inputString = "Hello C++ World";
7     std::istringstream iss(inputString);
8
9     std::string word;
10    while (iss >> word) {
11        // 处理每个单词，例如输出
12        std::cout << "Word: " << word << std::endl;
13    }
14
15    return 0;
16 }
```

这个方法使用了`std::istringstream`，它将输入字符串转换成类似于输入流的对象，然后通过`>>`运算符提取每个单词。

### 方法二：手动遍历字符

```
1 #include <iostream>
2 #include <string>
3
4 int main() {
5     std::string inputString = "Hello C++ World";
6
7     std::string word;
8     for (char c : inputString) {
9         if (c != ' ') {
10            word += c;
11        } else {
12            // 处理每个单词，例如输出
13            std::cout << "Word: " << word << std::endl;
14            word.clear();
15        }
16    }
17
18    // 处理最后一个单词
19    if (!word.empty()) {
20        std::cout << "Word: " << word << std::endl;
21    }
22
23    return 0;
24 }
```

这个方法通过手动遍历输入字符串中的字符，逐个构建单词，遇到空格则处理当前单词。

### 方法三：使用正则表达式

```
1 #include <iostream>
2 #include <regex>
3 #include <string>
4
5 int main() {
6     std::string inputString = "Hello C++ World";
7
8     std::regex wordRegex("\\S+"); // 匹配非空白字符的正则表达式
9
10    std::sregex_iterator it(inputString.begin(), inputString.end(), wordRegex);
11    std::sregex_iterator end;
12
13    while (it != end) {
14        // 处理每个匹配到的单词，例如输出
15        std::cout << "Word: " << it->str() << std::endl;
16        ++it;
17    }
18
19    return 0;
20 }
```

这个方法使用了正则表达式来匹配非空白字符，从而提取每个单词。

### 方法比较

这三种方法中，使用`std::istringstream`通常是最简单和效率较高的方法，因为它充分利用了C++标准库的功能。手动遍历字符的方法相对繁琐，而正则表达式方法可能会有一些性能开销，但在某些情况下更灵活。

选择方法应根据实际需求和性能要求来确定。如果只是简单地分割空格分隔的单词，`std::istringstream`是一个不错的选择。