

[C++]. Stringstream Trong C++ Và Ứng Dụng

<https://blog.28tech.com.vn/c-stringstream-trong-c-va-ung-dung>

Stringstream trong C++ là một công cụ hữu ích với các bài toán xử lý chuỗi ký tự, đặc biệt với các bài toán tách từ trong chuỗi. Bài học này mình sẽ hướng dẫn các bạn cách sử dụng stringstream để tách từ

NỘI DUNG :

- **Stringstream và bài toán tách từ**
- **Tách từ theo nhiều ký tự khác nhau**
- **Bài tập áp dụng stringstream**

1. Stringstream và bài toán tách từ

Thông thường khi chuẩn hóa hay xử lý chuỗi ký tự thì bạn cần tách riêng lẻ các từ trong chuỗi theo dấu cách hoặc một ký tự khác ví dụ như dấu . , ! ...

stringstream nằm trong thư viện **sstream**, nó có chức năng biến chuỗi string của bạn thành luồng (tương tự như luồng vào từ bàn phím là cin). Từ đó bạn có thể đọc từng từ trong luồng stringstream ra và xử lý.

Ví dụ sau mình sẽ tách từng từ trong chuỗi ra, sau khi tách xong bạn có thể chuẩn hóa, đếm từ ... với từ bạn tách được.

Mã nguồn 1:

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "28tech    hoc    lap trinh";
    //Khai báo stringstream ss và gán cho nó nội dung của s
    stringstream ss(s);
    string word;
    int dem = 0;
    while(ss >> word){
        ++dem;
        cout << "Tu thu " << dem << " tach duoc : " << word << endl;
    }
    return 0;
}
```

Output :

```
Tu thu 1 tach duoc : 28tech
Tu thu 2 tach duoc : hoc
Tu thu 3 tach duoc : lap
Tu thu 4 tach duoc : trinh
```

Giải thích mã nguồn :

- Ban đầu bạn khai báo 1 biến stringstream và khởi tạo giá trị cho nó bằng nội dung chuỗi bạn muốn tách từ
- Dùng toán tử >> để nhập 1 từ trong luồng stringstream cho biến word, vòng lặp kia sẽ lặp cho tới khi luồng stringstream của bạn không còn từ nào nữa thì nó sẽ dừng lặp.
- Giữa các từ có nhiều dấu cách sẽ được bỏ qua hết, tương tự như bạn nhập cin từ bàn phím thì số lượng dấu cách giữa các từ không có ý nghĩa.

Trong trường hợp bạn muốn tách theo 1 ký tự khác, ví dụ như là dấu - thay vì dấu cách như mặc định bạn có thể sử dụng hàm **getline()**, hoặc bạn có thể thay cách dấu - thành dấu cách và xử lý như tách từ bằng dấu cách.

Mã nguồn 2:

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "28tech-hoc-lap-trinh";
    //Khai báo stringstream ss và gán cho nó nội dung của s
    stringstream ss(s);
    string word;
    int dem = 0;
    while(getline(ss, word, '-')){
        ++dem;
        cout << "Tu thu " << dem << " tach duoc : " << word << endl;
    }
    return 0;
}
```

Output :

```
Tu thu 1 tach duoc : 28tech
Tu thu 2 tach duoc : hoc
Tu thu 3 tach duoc : lap
Tu thu 4 tach duoc : trinh
```

2. Tách từ theo nhiều ký tự khác nhau

Giả sử bạn cần tách các chuỗi theo 1 loạt ký tự khác nhau (delimiter) thì bạn nên chuyển hết các ký tự này thành dấu cách rồi tách chuỗi bằng stringstream.

Ví dụ muốn tách chuỗi theo các ký tự : dấu chấm '.', dấu phẩy ',', dấu cách ' ', dấu hỏi chấm '?', và dấu chấm than '!' ta triển khai như sau

Mã nguồn :

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "28tech,hoc.lap!!trinh???C++ dsa";
    //Thay hết delimiter bằng dấu cách
    for(int i = 0; i < s.size(); i++){
        if(s[i] == '?' || s[i] == '!' || s[i] == '.' || s[i] == ','){
            s[i] = ' ';
        }
    }
    //Khai báo stringstream ss và gán cho nó nội dung của s
    stringstream ss(s);
    string word;
    int dem = 0;
    while(ss >> word){
        ++dem;
        cout << "Tu thu " << dem << " tach duoc : " << word << endl;
    }
    return 0;
}
```

Output :

```
Tu thu 1 tach duoc : 28tech
Tu thu 2 tach duoc : hoc
Tu thu 3 tach duoc : lap
Tu thu 4 tach duoc : trinh
Tu thu 5 tach duoc : C++
Tu thu 6 tach duoc : dsa
```

3. Bài tập áp dụng stringstream

Bài tập 1 : Đếm số lượng từ trong xâu ký tự, giữa các từ có thể có nhiều dấu cách

Đối với bài toán này bạn có thể xử lý xâu bằng cách gặp ký tự khác dấu cách thì đếm 1 từ và xử lý nốt các ký tự còn lại của từ đó hoặc dùng stringstream

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "28tech 28tech com vn blog AI    python";
    stringstream ss(s);
    int cnt = 0;
    string word;
    while(ss >> word){
        ++cnt;
    }
    cout << "So luong tu trong xau : " << cnt << endl;
}
```

Output :

```
So luong tu trong xau : 7
```

Nếu bạn không muốn sử dụng stringstream thì có thể cài đặt như sau :

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main(){
    string s = "28tech 28tech com vn blog AI    python";
    int cnt = 0;
    for(int i = 0; i < s.size(); i++){
        if(s[i] != ' '){
            ++cnt;
            //duyet nốt các kí tự của từ bắt đầu từ chỉ số i
            while(i < s.size() && s[i] != ' '){
                ++i;
            }
            --i;
        }
    }
    cout << "So luong tu trong xau : " << cnt << endl;
}
```

Bài tập 2 : Chuẩn hóa từ và loại bỏ dấu cách thừa giữa các từ trong xâu, ví dụ " ngUYEN VaN tech28" thì chuẩn hóa thành "Nguyen Van Tech28"

Cách làm là bạn hãy tách các từ ra, chuẩn hóa từng từ rồi nối chung lại với nhau để tạo thành xâu chuẩn hóa. Thay vì bạn cứ loay hoay xét từ và xóa dấu cách thừa thì việc tách ra, chuẩn hóa và gộp lại sẽ dễ và nhanh hơn nhiều.

```
#include <iostream>
#include <string>
#include <ctype.h>
#include <sstream>

using namespace std;

int main(){
    string s = "    NgUYeN    vAN    tECH28    ";
    string res = "";
    stringstream ss(s);
    string word;
    while(ss >> word){
        //Ki tu dau tien cua tung tu thi viet hoa
        res += toupper(word[0]);
        //Cac ki tu con lai thi viet thuong
        for(int j = 1; j < word.size(); j++){
            res += tolower(word[j]);
        }
        //Them dau cach giua cac tu
        res += " ";
    }
    //Loai bo ki tu dau cach thua sau tu cuoi cung
    res.pop_back();
    cout << "Xau chuan hoa : " << res << endl;
}
```

Output :

Xau chuan hoa : Nguyen Van Tech28