

Stack Overflow em Português é um site de perguntas e respostas para programadores profissionais e entusiastas. Leva apenas um minuto para se inscrever.



Inscreva-se para participar desta comunidade

Qualquer pessoa pode fazer uma pergunta

Qualquer um pode responder

As melhores respostas recebem votos positivos e sobem para os primeiros lugares

Como separar uma string em pedaços?

Perguntada 3 anos, 5 meses atrás Ativa 3 meses atrás Vista 6mil vezes



7



Em outras linguagens existe o `split`, `explode` ou algo parecido que picota a *string* em pedaços de acordo com algum separador. Existe algo pronto em C ou tenho que fazer na mão?

c

c++

string

perguntada 17/03/17 às 11:20



Maniero ♦

413mil

64

1040

1568

2 Respostas

Ativos

Oldest

Votos



6



Não tem algo tão pronto, mas existe o `strtok()` que analisa a *string* e substitui um delimitador especificado por um caractere nulo e assim o que era uma única *string* vai virando várias, já que o nulo termina a *string* naquele ponto.

Mas note que ele não devolve um *array* de *strings* como é comum em outras linguagens, tão pouco faz com todos delimitadores. Ele faz apenas com o primeiro que encontrar, para fazer no segundo precisa roda o `strtok()` novamente e assim por diante. É claro que todo

Ao usar nosso site, você reconhece que leu e entendeu nossa Política de Cookies, Política de Privacidade e nossos Termos de Serviço.



```
#include <stdio.h>
#include <string.h>

int main(void) {
    char frutas[] = "banana,laranja,morango";
    int tamanho = strlen(frutas); //isto funciona só para delimitador de 1 caractere
    char *token = strtok(frutas, ",");
    for (int i = 0; i < tamanho; i++) printf(token[i] == 0 ? "\\0" : "%c", token[i]);
    while(token != NULL) {
        printf("\\n%s", token);
        token = strtok(NULL, ",");
    }
}
```

Veja [funcionando no ideone](#). E [no repl.it](#). Também [coloquei no GitHub](#) para referência futura.

editada 8/05 às 14:19

respondida 17/03/17 às 11:20



Maniero ♦

413mil 64 1040
1568



Em C++ não existe uma função nativa 'split' para strings.

1



Pesquisando o assunto encontra-se uma enorme variedade de maneiras de se separar uma string.



Alguns exemplos que achei interessante.

Exemplo 1

```
#include <iostream>
#include <sstream>
#include <string>
#include <vector>
using namespace std;

int main()
{
    // string a ser separada
    string tokenString { "aaa    bbb ccc" };

    // as sub-strings separadas vão ser colocadas neste vetor
    vector<string> tokens;

    // stream de strings de input inicializado com a string a ser separada
    istringstream tokenizer { tokenString };

    // variável de trabalho
    string token;

    // separa as string por espaço e coloca no vetor destino
    while (tokenizer >> token)
        tokens.push_back(token);
}
```



Resultado do exemplo 1:

```
* [aaa]
* [bbb]
* [ccc]
```

Exemplo 2

```
#include <iostream>
#include <sstream>
#include <string>
#include <vector>
using namespace std;

int main()
{
    // string a ser separada
    string tokenString { "aaa, bbb, ccc,,ddd  ,   eee" };

    // as sub-strings separadas vão ser colocadas neste vetor
    vector<string> tokens;

    // stream de strings de input inicializado com a string a ser separada
    istringstream tokenizer { tokenString };

    // variável de trabalho
    string token;

    // separa as sub-strings por vírgula e coloca no vetor destino
    while (getline(tokenizer, token, ','))
        tokens.push_back(token);

    // mostra na tela as sub-strings separadas
    for (const string& token: tokens)
        cout << "*" << "[" << token << "]\n";
}
```

Resultado do exemplo 2:

```
* [aaa]
* [ bbb]
* [ ccc]
* []
* [ddd  ]
* [   eee]
```

Notar que foram mantidos os espaços nas sub-strings destino. (Seria o caso de usar uma outra função comum para strings chamada 'trim' que *também* não existe em C++).

Exemplo 3

```
using namespace std;

int main()
{
    // string a ser separada
    string tokenString { "aaa, bbb, ccc,,ddd  ,   eee" };

    // as sub-strings separadas vão ser colocadas neste vetor
    vector<string> tokens;

    // expressão regular contendo os delimitadores: espaço e vírgula
    regex delimiters { "[\\s,]+" };

    // cria um iterator para um objeto contendo as sub-strings separadas
    // obs. estou usando uma "receita" pronta, não sei o motivo exato do parametro '-1'
    sregex_token_iterator tokens_begin { tokenString.begin(), tokenString.end(), delimiters,
-1 };

    // iterator finalizador
    auto tokens_end = sregex_token_iterator {};

    // copia as sub-strings separadas para o vetor destino
    for (auto token_it = tokens_begin; token_it != tokens_end; token_it++)
        tokens.push_back(*token_it);

    // mostra na tela as sub-strings separadas
    for (const string& token: tokens)
        cout << "*" << token << "]\n";
}
```

Resultado do exemplo 3:

```
* [aaa]
* [bbb]
* [ccc]
* [ddd]
* [eee]
```

That's all for now folks.

editada 11/06 às 14:45



Comunidade ♦

1

respondida 18/03/17 às 23:24



zentrunix

5.395

2

12

21