

OS ARGUMENTOS `argc` e `argv`

A função `main()` é a função obrigatória de qualquer programa escrito em C++. Normalmente, apresentamos sem parâmetros. No entanto, ela pode receber parâmetros formais.

```
int main (int argc, char *argv[])
```

- **argc** (*argument count*) O parâmetro `argc` é um inteiro e contém o número de argumentos com os quais o programa foi chamado na linha de comando. Ele é no mínimo 1, pois o nome do programa é contado como sendo o primeiro argumento.
- **argv** (*argument values*) O parâmetro `argv` é um ponteiro para uma matriz de strings. Cada string dessa matriz é um dos parâmetros da linha de comando. `argv[0]` sempre aponta para o nome do programa (que, como já foi dito, é considerado o primeiro argumento). É para saber quantos elementos temos em `argv` que temos `argc`.

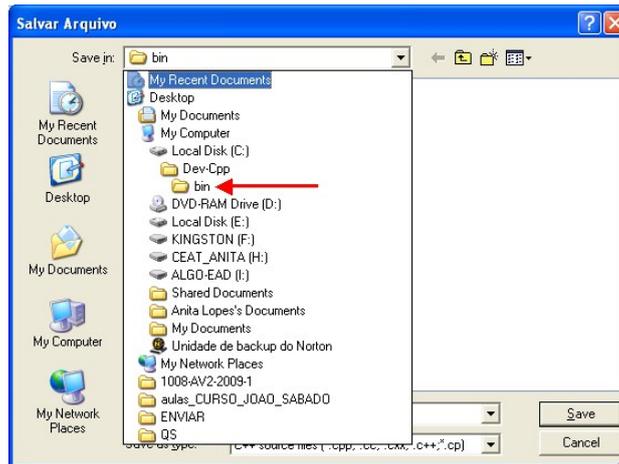
Os nomes dos parâmetros "`argc`" e "`argv`" podem ser alterados, mas, normalmente, talvez por questão de padronização, não se costuma mudá-los.

Para que posamos executar os programas abaixo, você terá que executá-los no terminal e já que pode ser uma novidade para alguns, vou orientá-los passo a passo.

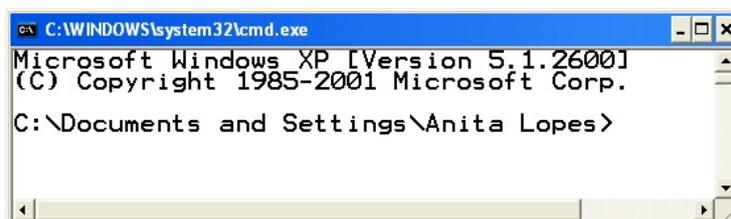
1) Digite seu arquivo:

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    if(argc!=3)
    {
        cout<<"Voce nao digitou dois nomes\n";
        exit (1);
    }
    cout<<"Ola " <<argv[1]<<" e " <<argv[2]<<"\n\n";
    system("pause");
}
```

Salve seu arquivo na pasta **bin** dentro da pasta **Dev-Cpp**, isto é: `Dev-Cpp\bin`.



2) Ao clicar em iniciar, aparecerá a opção executar. Digite: **cmd** e enter

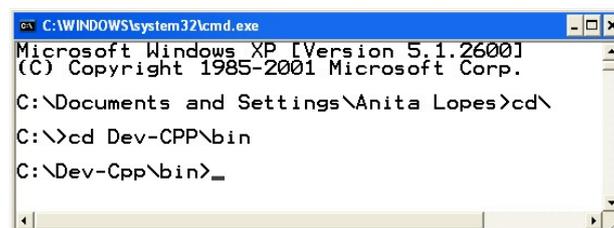


3) Sei que você é da geração do mouse, mas esqueça dele por alguns instantes e use o teclado.

Você agora irá para a raiz do drive C, isto é, trocará de diretório(change directory).



4) Vamos agora, de uma só vez, mudar para um diretório que está dentro de outro diretório.



5) Estando no diretório onde se encontra o executável *arg.exe*, vamos então, executá-lo com seus parâmetros.

Como alguns de você nunca devem ter feito isso antes, é bom dizer que, para executar um programa no terminal, temos que digitar o nome do programa e pressionar a tecla enter.

Entretanto, nesse caso precisaremos digitar o nome do programa e mais duas strings porque estão sendo esperados três argumentos. Observe o teste: `if(argc!=3)`

```

C:\WINDOWS\system32\cmd.exe - arg ANITA JOAO
C:\Dev-Cpp\bin>arg ANITA JOAO
Ola ANITA e JOAO
Press any key to continue . . .
  
```

Mais um exemplo com um número variado de elementos

Arquivo

```

#include <iostream>
using namespace std;
int main( int argc, char * argv[] )
{
    int x;
    cout << "\nNome do programa : " << argv[ 0 ];
    cout << "\nDemais nomes\n";
    //argc conta os demais argumentos na entrada
    for ( x= 1; x < argc; x++)
        cout << "\n" << x << " - " << argv[x];
    cout<<"\n\n";
    system("pause");
    return 0;
}
  
```

```

C:\WINDOWS\system32\cmd.exe - arg1 ANITA JOAO
C:\Dev-Cpp\bin>arg1
Nome do programa : arg1
Demais nomes

Press any key to continue . . .
C:\Dev-Cpp\bin>arg1 ANITA
Nome do programa : arg1
Demais nomes

1 - ANITA
Press any key to continue . . .
C:\Dev-Cpp\bin>arg1 ANITA JOAO
Nome do programa : arg1
Demais nomes

1 - ANITA
2 - JOAO
Press any key to continue . . .
  
```



Observe que na primeira vez, só digitamos o nome do programa. Na segunda, o nome do programa e um outro argumento e, na terceira, o nome do programa e dois argumentos. O argumento **argc** contém o número de elementos da matriz **argv []**, incluindo o nome do programa.

É uma prática muito comum usar a função *main()* sem argumentos, mas como afirma Schildt, H(1996) "O uso dos argumentos da linha de comando dá aos seus programas uma aparência profissional e facilita o uso do programa em arquivos de lote".

É claro que esses argumentos não servem só para isso, mas gostaria de despertar em vocês algo que pudesse fazer uma ponte com a disciplina de Sistemas Operacionais.