

Argumentos de línea de comando en C / C ++

Todos sabemos que la función más importante de cualquier programa es el main. Generalmente se define sin parámetros y con un tipo de retorno a entero.

```
int main()
{
/* ... */
}
```

Pero También podemos tener argumentos de línea de comandos tanto en C como C ++. Los argumentos de la línea de comandos se ponen después del nombre del programa en la línea de comandos.

Se usan principalmente para controlar el programa desde afuera en lugar de poner esos valores dentro del código. Por ejemplo, si hacen un juego pueden pasarle por comando la cantidad de jugadores y a partir de eso correr distintas versiones del programa. O bien pueden pasarle el nombre de un archivo de donde quieren que lea o escriba los datos.

Los argumentos se pasan a la funcion main definiendo dos argumentos:

El primer argumento es el número de argumentos de la línea de comandos y el segundo es la lista de argumentos de la línea de comandos. Pueden definirse de la siguiente forma:

```
int main(int argc, char *argv[]) { /* ... */ }
```

o

```
int main(int argc, char **argv) { /* ... */ }
```

Estas dos formas son ambas validas porque es lo mismo escribir `*argv[]` que `**argv`, porque en definitiva un vector es un puntero a la primer posición del vector. Esto lo van a ver mejor cuando tengan la clase de punteros. Entonces si tengo un puntero a un vector, es lo mismo que tener un puntero a un puntero (doble puntero).

- **argc (ARGument Count)** es de tipo int y guarda el número de argumentos pasados, incluyendo el nombre del programa. Entonces, si le pasamos un valor a un argumento al programa cuando lo corremos desde la terminal, el valor de argc va a ser 2 (uno para el argumento y otro para el nombre del programa)

- **argv(Argument Vector)** es una matriz de punteros a char que enumera todos los argumentos.
 - Si argc es mayor que cero, los elementos de la matriz de argv [0] a argv [argc-1] van a contener punteros a cadenas.
 - argv [0] es el nombre del programa, después de eso hasta argv [argc-1] cada elemento son argumentos de línea de comandos.
 - argv [argc] es un puntero NULL.
 - argv [0] contiene el nombre del programa.
 - argv [1] apunta al primer argumento de la línea de comando y argv [n] señala el último argumento.

Estos parámetros que mencionamos se pasan al programa cuando se invoca desde la terminal. Veamos un ejemplo:

```
// Nombre del programa: main.cpp

#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{
    cout << "Ingresaste " << argc << " argumentos:" << "\n";

    for (int i = 0; i < argc; ++i)
        cout << argv[i] << "\n";

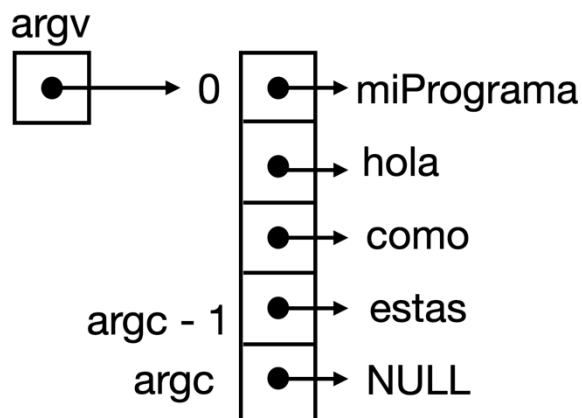
    return 0;
}
```

Input:

```
$ g++ main.cpp -o programa
$ ./programa hola como estas
```

Output:

```
Usted ingreso 4 argumentos:
./miPrograma
Hola
Como
estas
```



Nota: Se pasan todos los argumentos de la línea de comandos separados por un espacio, pero si el argumento en sí tiene un espacio, se pueden pasar los argumentos poniéndolos entre comillas dobles "" o comillas simples ".

Veamos otro ejemplo:

```

int main(int argc, char** argv)
{
    int i;
    cout<<"El nombre del programa es" << argv[0]<<endl;

    if(argc==1)
        cout<<"No hay comando de linea extra pasado mas que el
nombre del programa"<<endl;
    if(argc>=2)
    {
        cout<<"Numero de argumentos pasados:" << argc<<endl;
        cout<<"Los argumentos pasados fueron: ";
        for(i=0;i<argc;i++)
            cout<<"argv["<<i<<"]: "<<argv[i]<<endl;
    }
    return 0;
}

```

Analicemos la salida para distintos casos:

- Sin argumentos

```
./programa
```

El nombre del programa es ./programa

No hay comando de linea extra pasado mas que el nombre del programa

- Tres argumentos

```
./programa hola como estas
```

El nombre del programa es ./programa

Numero de argumentos pasados: 4

Los argumentos pasados fueron:

argv[0]: ./programa

argv[1]: hola

argv[2]: como

argv[3]: estas

- Argumento simple: Se ejecuta con un solo argumento separado por espacio pero dentro de comillas dobles o simples

```
./programa "hola como estas"
```

Numero de argumentos pasados: 2

Los argumentos pasados fueron:

argv[0]: ./programa

argv[1]: hola como estas