Command line parameter in C

This chapter only really makes sense for you if you are using a command promt to compile the program. It is possible to pass values ??from the command line - command line to program C when it is executed.

This chapter only really makes sense for you if you are using a **command promt** to compile the program. It is possible to pass values ??from the command line - command line to program C when it is executed. These values ??are called **Command line arguments** and are sometimes important for your program when you control your program outside instead of raw coding values ??inside the code.

The command line parameters are handled using the **main ()** function parameters, with **argc** pointing to the number of parameters you pass and **argv []** as the pointer array to any parameter provided for that program . Here is an example to check if there are any parameters provided from the command line and perform the corresponding actions:

```
#include int  main ( int  argc , char * argv [] ) { if (  argc  == 2 ) {  print
```

When the above code is compiled and executed with a parameter, it will print the following result:

```
$ ./ a . out  thamso1  Tham  so duoc cung cap la :  thamso1
```

When you pass two parameters to the code above, it will print the following result:

```
$ ./ a . out  thamso1 thamso2  Qua  nhieu tham so duoc cung cap .
```

When the above code is executed and executed with no parameters passed, it will print the result below:

```
$ ./ a . out Ban  nen cung cap mot tham so .
```

Note that **argv [0]** holds the name of the program itself and **argv [1]** is a pointer to the first command line parameter provided, **argv [n]** is the last parameter. If no parameters are provided, **argc** will be 1, if you pass a parameter then **argc will** have a value of 2.

You pass all command line parameters separately by a space, but if the parameters itself have a space, you can pass these parameters by placing them in double quotation marks ("") or quoting single ('). Now we rewrite the above program when you print out the program name and pass the command line parameters inside the double quotation mark ("").

```
#include int  main ( int  argc , char * argv [] ) {  printf ( "Ten chuong trinh
```

When the above code is compiled and executed with a single parameter by a space inside the double quotation mark, the following result is printed:

```
$ ./ a . out "thamso1 thamso2" Ten  chuong trinh la : ./ a . out Tham  so duoc
```

According to Tutorialspoint

Previous lesson: Memory management in C

Next article: What is C ++?

You finished reading the article "**Command line parameter in C**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
