**C语言命令行参数的使用**



[luoganttcc](https://blog.csdn.net/luoganttcc) 2021-01-14 17:54:58  22  收藏

分类专栏： [c语言](https://blog.csdn.net/luoganttcc/category_7610235.html)

版权

[原文链接](https://blog.csdn.net/xiaocainiaoshangxiao/article/details/13996459#commentBox)

**之前曾经使用过很多次c语言的命令行参数了，但是总是每次使用的时候都不太确定，需要重新查资料，这次来个总结。c语言的命令行参数非常简单，只需要一个简单的例子就可以说明**

**新建test.c**

//文件名称是 test.c

#include <stdio.h>

void main(int argc,char\*\* argv)

{

printf("%d\n",argc);

printf("%s\n",argv[0]);

printf("%s\n",argv[1]);

printf("%s\n",argv[2]);

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

**在上面的例子中，我们给main函数传递两个参数，argc和argv。argc是int类型的，它表示的是命令行参数的个数。不许要用户传递，它会根据用户从命令行输入的参数个数，自动确定。argv是char\*\*类型的，它的作用是存储用户从命令行传递进来的参数。它的第一个成员是用户运行的程序名字。**

**对于上面的例子，我们将其保存为test.c，用gcc编译生成目标文件为test，测试如下：**

（1）我们值在命令行下直接运行程序，没有传递其它任何参数：

gcc test.c -o test

* 1

./test

* 1

**第一个输出的是argc，因为我们只输入了./test，所以argc为1，即只有一个命令行参数。后面输出的第一个命令行参数也是./test。接着程序就出错了，因为后面已经没有第二个和第三个参数了，在写实际的应用时，应注意对这一部分进行容错处理。**

1

./test

段错误 (核心已转储)

* 1
* 2
* 3

（2）我们在命令行下输入：

./test hello world

* 1

3

./test

hello

world

* 1
* 2
* 3
* 4

（3）我们继续输入：

./test hello world hello world

* 1

**可以看到这时，argc变为5，argv指数出了前面三个参数，当然还可以把后面的两个参数也进行输出。**

5

./test

hello

world

* 1
* 2
* 3
* 4

如果命令行中传递的一个参数包括空格，就需要用用“ ”将参数扩起来例如：

* 1

./test "hello world" "hello world"

* 1

3

./test

hello world

hello world

* 1
* 2
* 3
* 4

我们将hello world整体当作一个参数传递，就需要使用“hello world”。

**需要注意的是argv的类型可以是char \*\* argv，char\* argv[ ]。**