

C\_C 中命令行参数的原理 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_C\\_C\\_\\_\\_E4\\_B8\\_AD\\_E5\\_91\\_c97\\_133848.htm](https://www.100test.com/kao_ti2020/133/2021_2022_C_C___E4_B8_AD_E5_91_c97_133848.htm)

很早以前的dos应用程序中，我们经常会用命令行参数执行应用程序例如：`copy c:\1.txt d:\`在c/c中，命令行参数的传递是利用main进行形参传递实现的。前面的教程中我们对面都进行了void main()或者是void main(void)的定义,这表示main不返回任何类型,和main不带有任何参数，但在缺剩状态下mian()这样形式下，形式参数一直是存在在栈空间内的只是隐藏了起来。为了实现命令行参数我们将使用main(int argc,char\* argv[])这样的形式进行定义argc和argv可以换成你自己喜欢的名称，不一定要用argv，argc这些形式只是习惯而已，char\* argv[]我们前面已经讲述过，这就是一个指向指针数组，argv就是一个指针数组名，argv不是常量指针，而是具备变量特性的变量指针，它是可以移动的，由此我们可以改写成char\* \*argv也是正确的，int argc这个定义返回的将是参数的个数所以标记为整形(int)。

```
#include <string.h>
#include <iostream>
using namespace std. void main(int argc,char* argv[]) { int i=0. while(i { cout << argv[i] << endl; cin.get(). }
```

在命令行参数的提交中，系统会自动给指针数组后加上一个NULL，所以代码可以写成如下简单形式，简化代码提高效率。

```
#include <string.h>
#include <iostream>
using namespace std. void main(int argc,char* argv[]) { while(*argv!=NULL)//这里写成while(*argv)也是正确的! { cout << argv[i] << endl; cin.get(). }
```

最后我们说一下完整的命令行参数的原形：`void main(int argc,char* argv[],char* env[])` env用的很少，通常是用做事件的返回，这里就不讨论了，具体的可以查询书籍。

void main(int argc)省略其它参数的定义也是可以的，这样运行时候就直接返回参数个数，而不返回其它。运行命令行参数带有char\* argv[]的时候，如果输入参数带有空格式，应该用双引号括起来。例如 test4.exe "hello world!" ha ha 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)