

argc和argv

2020年11月7日 7:37

一.main()函数参数

通常我们在写主函数时都是void main()或int main() {..return 0;}，

但ANSI-C（美国国家标准协会,C的第一个标准ANSI发布）在C89/C99中main()函数主要形式为：

(1).int main(void)

(2).int main(int argc,char *argv[])

=

int main(int argc,char **argv).

其参数argc和argv用于运行时,把命令行参数传入主程序.其中ARG是指arguments,即参数.具体含义如下:

(1).int argc:英文名为arguments count(参数计数)

count of cmd line args,运行程序传送给main函数的命令行参数总个数,包括可执行程序名,其中当argc=1时表示只有一个程序名称,此时存储在argv[0]中.

(2).char **argv:英文名为arguments value/vector(参数值)

pointer to table of cmd line args,字符串数组,用来存放指向字符串参数的指针数组,每个元素指向一个参数,空格分隔参数,其长度为argc.数组下标从0开始,argv[argc]=NULL.

argv[0] 指向程序运行时的全路径名

argv[1] 指向程序在DOS命令中执行程序名后的第一个字符串

argv[2] 指向执行程序名后的第二个字符串

argv[argc] 为NULL.

二.源代码中的argc与argv

C程序必须有main()函数为入口,而且它不能被其他函数调用(可以调用自身),因此不能再程序内部取得实际值.

何处把实参赋值给main函数的形参呢? 这就需要调用"运行"或"DOS提示符",在调用可执行程序exe时,编译器会帮助我们将输入参数的信息放入main函数的参数列表中传参.

• argc、argv的具体含义

argc和argv参数在用命令行编译程序时有用。main(int argc, char* argv[], char **env) 中第一个参数，int型的argc，为整型，用来统计程序运行时发送给main函数的命令行参数的个数，在VS中默认值为1。

第二个参数，char*型的argv[]，为字符串数组，用来存放指向的字符串参数的指针数组，每一个元素指向一个参数。各成员含义如下：

argv[0]指向程序运行的全路径名

argv[1]指向在DOS命令行中执行程序名后的第一个字符串

argv[2]指向执行程序名后的第二个字符串

argv[3]指向执行程序名后的第三个字符串

argv[argc]为NULL

第三个参数，char**型的env，为字符串数组。env[]的每一个元素都包含ENVVAR=value形式的字符串，其中ENVVAR为环境变量，value为其对应的值。平时使用到的比较少。

- 计算命令行参数个数程序如下:

```
<strong>//C 输出参数个数
#include <stdio.h>
#include <stdlib.h>
int main(int argc,char *argv[])
{
printf("参数个数=%d\n",argc);
system("PAUSE");
return 0;
}

//C++ 输出参数个数
#include <iostream>
using namespace std;
int main(int argc,char *argv[])
{
cout<<"参数个数="<<argc<<endl;
system("PAUSE");
return 0;
}</strong>
```

调用"运行"(快捷键Ctrl+R)或"cmd"输入"G:\test.exe"会输出"参数个数=1",此时存储的就是执行程序名.输入"G:\test.exe 2 hello good"输出"参数个数=4":

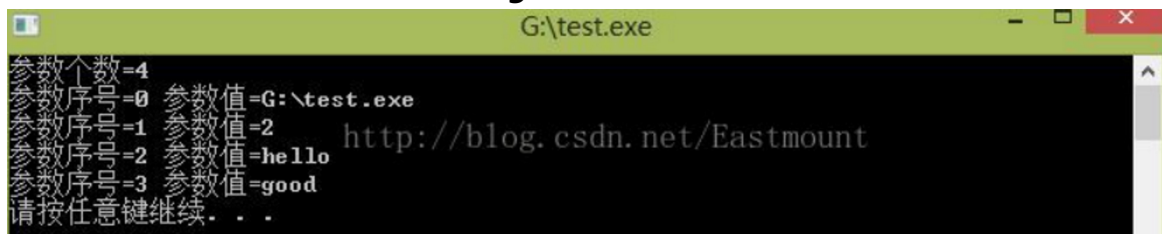


- 查看argv[]二维数组存储具体字符串的代码如下:

```
<strong>//C 查看argv存储参数值及对应序号
#include <stdio.h>
#include <stdlib.h>
int main(int argc,char *argv[])
{
int i;
printf("参数个数=%d\n",argc);
for(i=0; i<argc; i++)
{
printf("参数序号=%d ",i);
```

```
printf("参数值=%s\n",argv[i]);
}
system("PAUSE");
return 0;
}
//C++ 查看argv存储参数值及对应序号
#include <iostream>
using namespace std;
int main(int argc,char *argv[])
{
cout<<"参数个数="<<argc<<endl;
for(int i=0; i<argc; i++)
{
cout<<"参数序号="<<i<<" ";
cout<<"参数值="<<argv[i]<<endl;
}
system("PAUSE");
return 0;
}
```

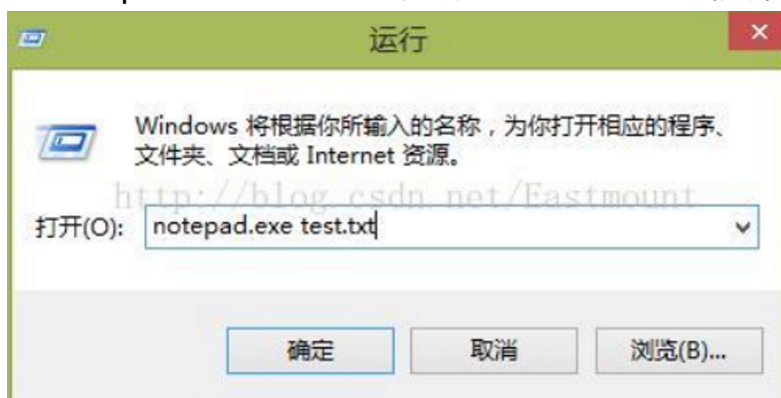
"运行"中输入"G:\test.exe 2 hello good",则输出如下图所示:



其中argv[0]指向字符串可执行程序的名称G盘下的test.exe,通常会位于"项目名称\Debug\xxx.exe"中.后面argv[1..3]单元依次指向程序调用时的参数.

• 三.调用argc和argv

在"运行"中输入"notepad.exe"回车能执行记事本程序(位于C:\Windows\System32),如果输入"notepad.exe test.txt"可以打开test.txt文本文件,其中test.txt位于当前路径下.如下图所示:



,main()函数其实与之也类似.同时在使用文件知识时,我们通常会涉及到main函数的argc和argv参数.如在《C++ Primer》这本书中第10.3.9实现单词转换的例子就涉及到该运用,这里只讲述涉及到该参数的部分代码供大家参考,大家可以自己去学习了解:

```
#include <iostream>
using namespace std;
```

```
int main(int argc, char *argv[])
{
//参数个数=3:工程名\读取txt文件\写入txt文件
if(argc!=3)
throw runtime_error("wrong number of arguments");
//打开转换文件,argv[1]为读取的要转换txt文件名 open_file自定义打开函数
if(!open_file(map_file,argv[1]))
throw runtime_error("no transformation file");
//打开要转换的写入txt文件
if(!open_file(input,argv[2]))
throw runtime_error("no input file");
return 0;
}
```