



# 程序设计实习

郭炜 微博 <http://weibo.com/guoweiofpku>

<http://blog.sina.com.cn/u/3266490431>

刘家瑛 微博 <http://weibo.com/pkuliujiaying>

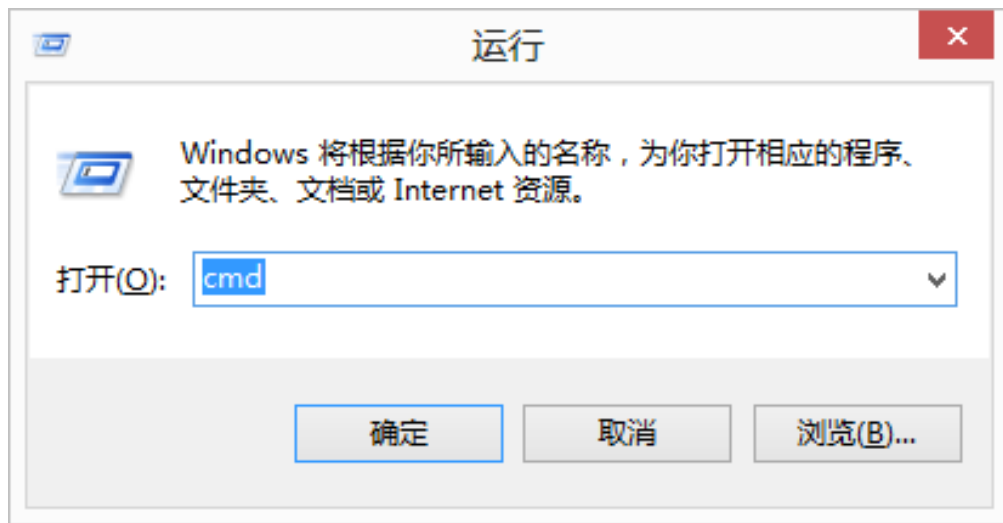


# 命令行参数

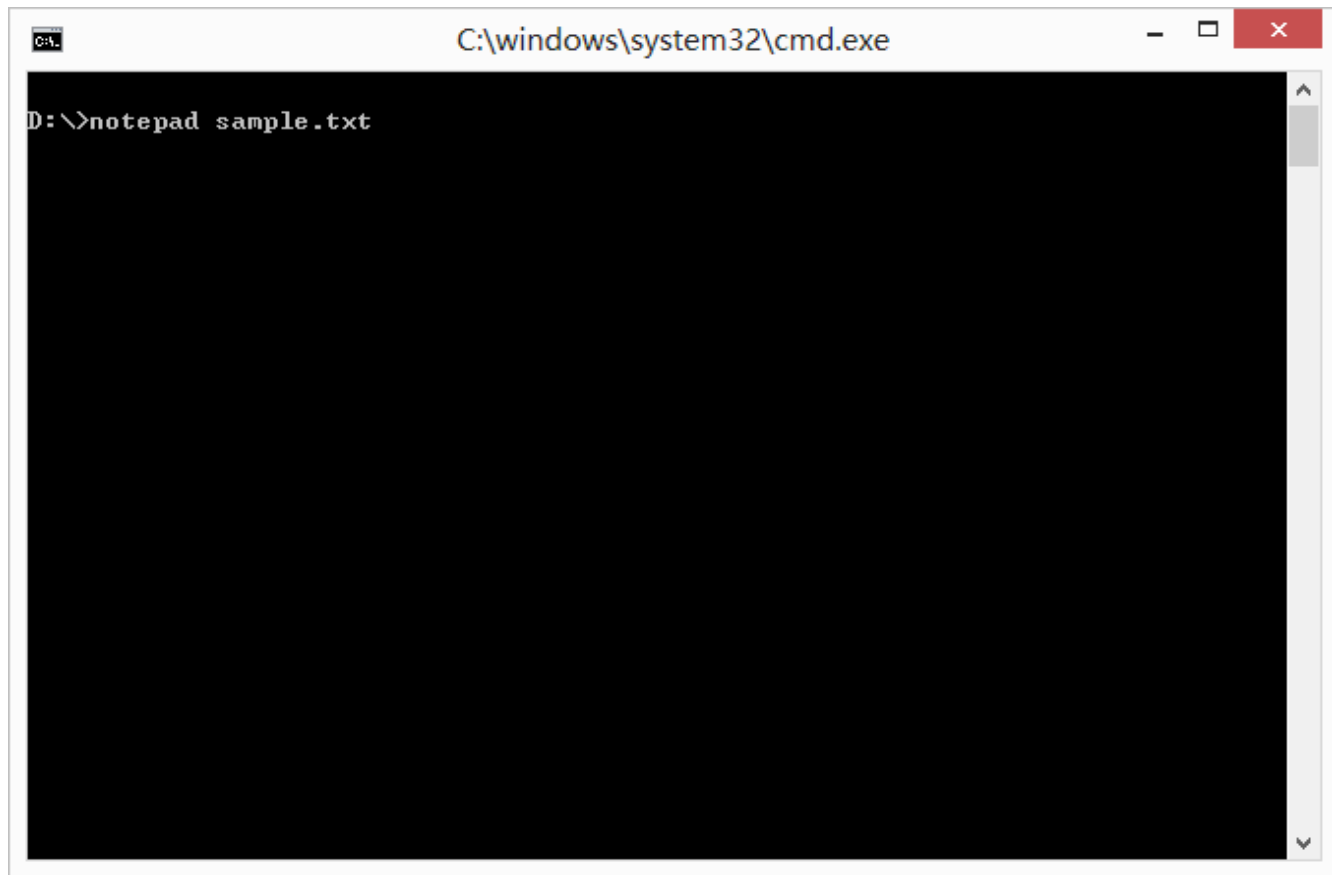
(教材P157)

# 命令行方式运行程序

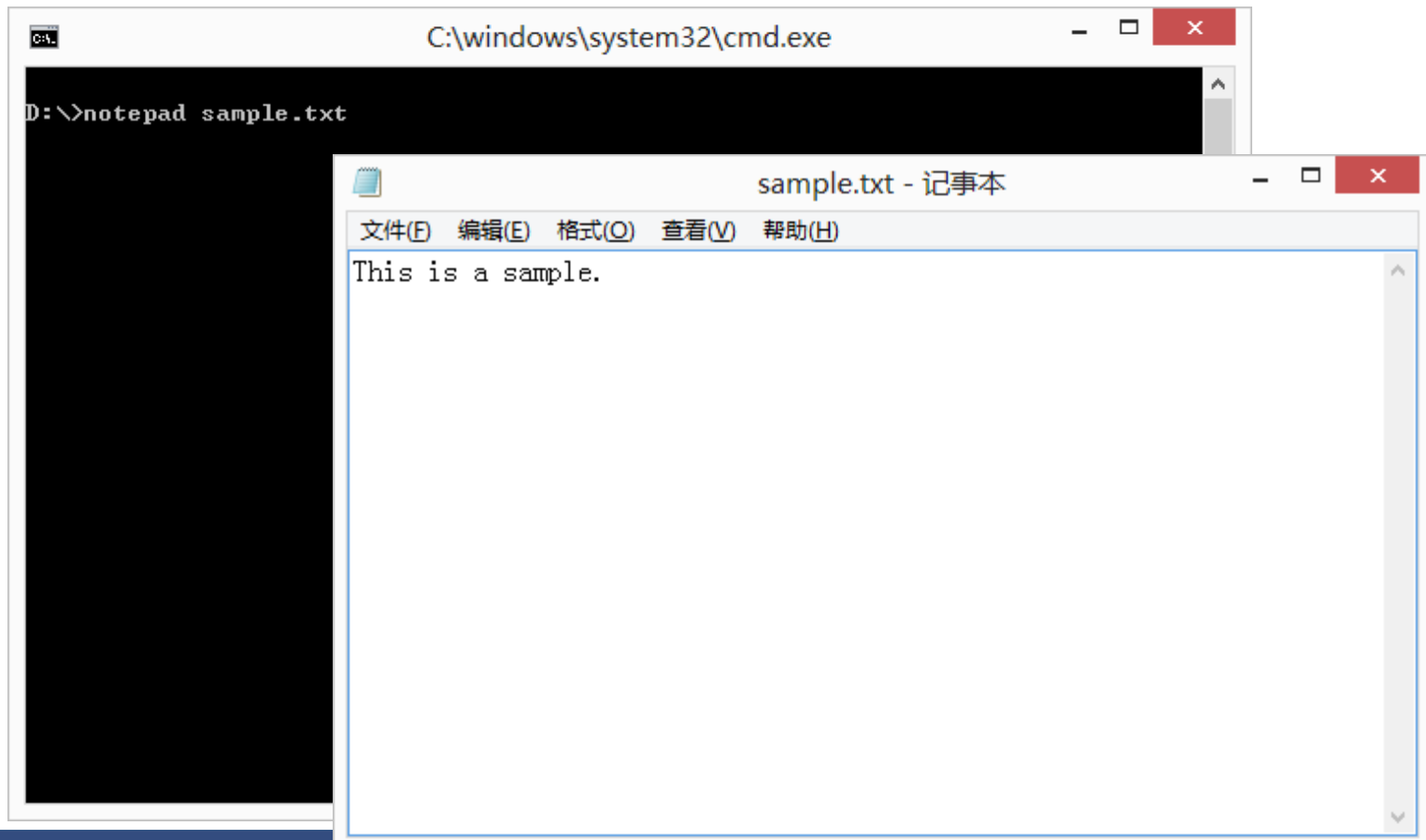
Windows + R 键:



# 命令行方式运行程序



# 命令行方式运行程序



# 命令行方式运行程序

```
notepad sample.txt
```

notepad 程序如何得知，用户在以命令行方式运行它的时候，后面跟着什么参数？

# 命令行参数

将用户在CMD窗口输入可执行文件名的方式启动程序时，跟在可执行文件名后面的那些字符串，称为“命令行参数”。命令行参数可以有多个，以空格分隔。比如，在CMD窗口敲：

```
copy file1.txt file2.txt
```

“copy”，“file1.txt”，“file2.txt”  
就是命令行参数

如何在程序中获得命令行参数呢？

# 命令行参数

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

**argc:** 代表启动程序时，命令行参数的个数。C/C++语言规定，可执行程序程序本身的文件名，也算一个命令行参数，因此，argc的值至少是1。



# 命令行参数

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

**argc:** 代表启动程序时，命令行参数的个数。C/C++语言规定，可执行程序程序本身的文件名，也算一个命令行参数，因此，argc的值至少是1。

**argv:** 指针数组，其中的每个元素都是一个char\* 类型的指针，该指针指向一个字符串，这个字符串里就存放着命令行参数。

例如，argv[0]指向的字符串就是第一个命令行参数，即可执行程序的文件名，argv[1]指向第二个命令行参数，argv[2]指向第三个命令行参数……。

```
#include <stdio.h>
int main(int argc, char * argv[])
{
    for(int i = 0; i < argc; i ++ )
        printf( "%s\n", argv[i]);
    return 0;
}
```

将上面的程序编译成sample.exe，然后在控制台窗口敲：

```
sample para1 para2 s.txt 5 "hello world"
```

```
#include <stdio.h>
int main(int argc, char * argv[])
{
    for(int i = 0; i < argc; i ++ )
        printf( "%s\n", argv[i]);
    return 0;
}
```

将上面的程序编译成sample.exe，然后在控制台窗口敲：

sample para1 para2 s.txt 5 “hello world”

输出结果就是：

sample  
para1  
para2  
s.txt  
5  
hello world