

命令行参数

- 什么是**命令行参数**（**Command Line Arguments**）？
- GUI界面之前，计算机的操作界面都是字符式的命令行界面（DOS、UNIX、Linux）
- 例如，在DOS下拷贝文件用
 - * `copy file1.c file2.c`
- 不仅是操作系统命令，所有程序也都有命令行信息



命令行参数

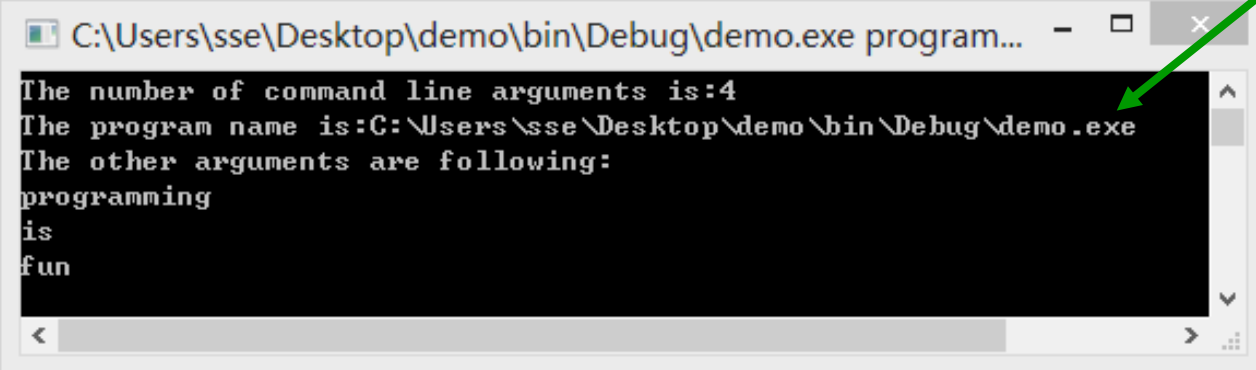
- 命令行参数有什么用？
 - * 使用户可以根据需要来决定程序干什么、怎么干
- 如何访问命令行参数？
- `int main(int argc, char* argv[])`
 - * `argc`: 命令行参数的数量（包括程序名本身）
 - * `argv`: 指向命令行参数的指针数组
 - * `argv[0]` 为指向程序名的字符指针
 - * `argv[1]~argv[argc-1]` 为指向余下的命令行参数的字符指针



演示命令行参数与main函数形参间的关系

```
int main(int argc, char *argv[])
{
    int i;
    printf("The number of command line arguments is:%d\n", argc);
    printf("The program name is:%s\n", argv[0]);
    if (argc > 1)
    {
        printf("The other arguments are following:\n");
        for (i = 1; i < argc; i++)
        {
            printf("%s\n", argv[i]);
        }
    }
    return 0;
}
```

在不同的操作系统上
程序名可能会包含
路径或其他信息



```
C:\Users\sse\Desktop\demo\bin\Debug\demo.exe program...
The number of command line arguments is:4
The program name is:C:\Users\sse\Desktop\demo\bin\Debug\demo.exe
The other arguments are following:
programming
is
fun
```

指针数组

argv[0]

字符串

程序名

argv[1]

programming

argv[2]

is

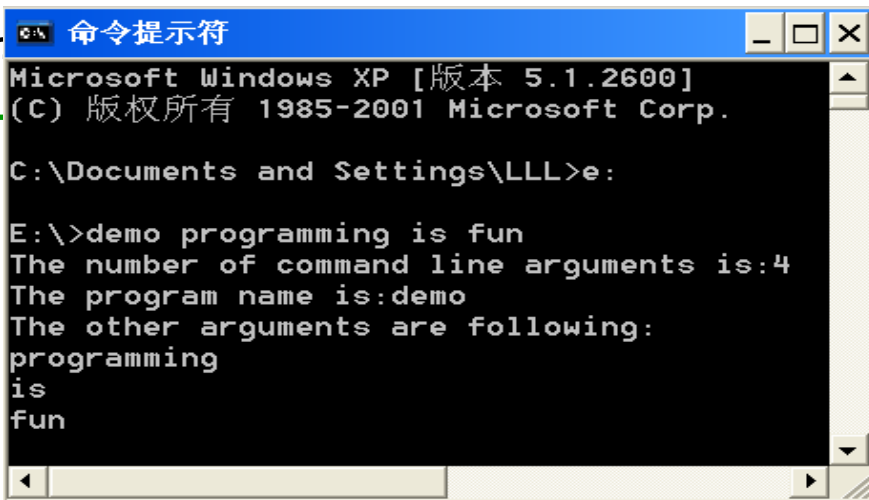
argv[3]

fun

演示命令行参数与main函数形参间的关系

```
int main(int argc, char *argv[])
{
    int i;
    printf("The number of command line arguments is:%d\n", argc);
    printf("The program name is:%s\n", argv[0]);
    if (argc > 1)
    {
        printf("The other arguments are following:\n");
        for (i = 1; i < argc; i++)
        {
            printf("%s\n", argv[i]);
        }
    }
    return 0;
}
```

如何输入命令行参数?
demo.exe programming is fun



命令提示符

Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\LLL>e:

E:\>demo programming is fun

The number of command line arguments is:4

The program name is:demo

The other arguments are following:

programming

is

fun

