

第6章 函数

——断言与防御式编程

如何确定假设的真假？

■ 程序中的假设

- * 某个特定点的某个表达式的值一定**为真**
- * 某个特定点的某个表达式的值一定**位于某个区间等**

■ 问题：如何确定这些假设的**真假**呢？

■ 断言（Assert）

- * 测试程序中假设的正确性
- * 如果假设被违反，则中断程序的执行



断言

■ 在<assert.h>中定义的宏

```
#include <stdio.h>
#include <assert.h>
unsigned long Fact(unsigned int n);
int main()
{
    int m;
    do{
        printf("Input m(m>=0):");
        scanf("%d", &m);
    }while (m < 0);
    assert(m >= 0);
    printf("%d! = %lu\n", m, Fact(m));
    return 0;
}
```

`void assert(int expression);`
expression为真，无声无息
expression为假，中断程序

```
unsigned long Fact(unsigned int n)
{
    unsigned int i;
    unsigned long result = 1;
    for (i=2; i<=n; i++)
    {
        result *= i;
    }
    return result;
}
```

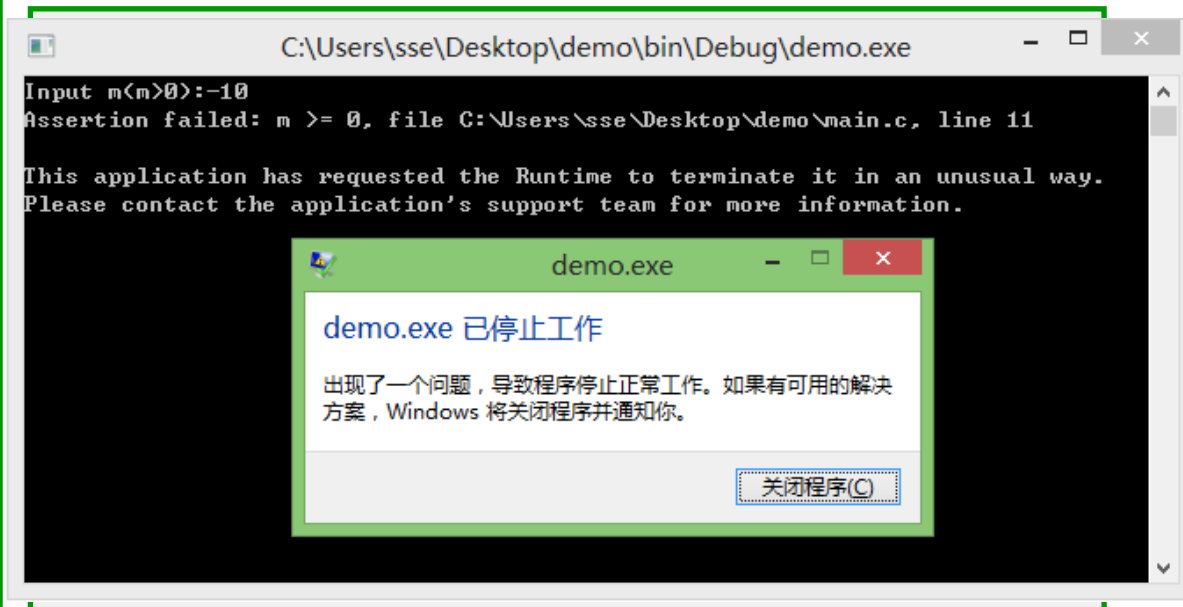
断言

■ 在<assert.h>中定义的宏

```
#include <stdio.h>
#include <assert.h>
unsigned long Fact(unsigned int n);
int main()
{
    int m;
    do{
        printf("Input m(m>=0):");
        scanf("%d", &m);
    }while (m >= 0);
    assert(m >= 0);
    printf("%d! = %lu\n", m, Fact(m));
    return 0;
}
```

`void assert(int expression);`
expression为真，无声无息
expression为假，中断程序

```
if (m < 0) exit(1);
```



断言

- 问题：使用条件语句代替断言，可不可以？
 - 使用断言便于在调试程序时发现错误，不会影响程序执行效率
- 仅用于调试程序，不能作为程序的功能

```
#include <stdio.h>
#include <assert.h>
long Fact(int n);
int main()
{
    int m;
    printf("Input m(m>=0):");
    scanf("%d", &m);
    assert(m >= 0);
    printf("%d! = %lu\n", m, Fact(m));
    return 0;
}
```

```
long Fact(int n)
{
    int i;
    long result = 1;
    assert(n >= 0);
    for (i=2; i<=n; i++)
    {
        result *= i;
    }
    return result;
}
```

断言

- 问题：使用条件语句代替断言，可不可以？
 - 使用断言便于在调试程序时发现错误，不会影响程序执行效率
- 仅用于调试程序，不能作为程序的功能

```
int MakeNumber(void)
{
    int number;
    number = rand() % 100 + 1;
    assert(number >= 1 && number <= 100);
    return number;
}
```



断言

- 何时适合使用断言呢？
 - * 检查程序中的各种假设的正确性
 - * 证实或测试某种不可能发生的状况确实不会发生
- 使用断言的基本原则
 - * 使用断言捕获不应该或者不可能发生的情况
 - * 每个assert只检验一个条件

防御式编程（Defensive programming）

- 让你编写的代码具有防弹功能



防御式编程（Defensive programming）

- 养成良好的编码风格
- 避免闪电式编程，用怀疑的眼光审视所有的输入和结果
 - * If anything can go wrong, it will——Murphy's Law
- 简单就是一种美，不要滥用技巧，让你的代码过于复杂
- 编译时打开所有警告开关，不要忽略它们
- 使用安全的数据结构和函数调用
- 做内存的“好管家”

讨论

- 你对防御式编程怎么看？你听说过墨菲定律吗？你怎样理解墨菲定律？你认为防御式编程与墨菲定律这二者有关系吗？



断言

- 问题：使用条件语句代替断言，可不可以？
 - 使用断言便于在调试程序时发现错误，不会影响程序执行效率
- Debug版本
 - 用于内部调试 —— assert是仅在Debug版本中起作用
- Release版本
 - 发行给用户使用 —— 编译器会跳过assert不生成检查代码

