

# 操作系统相关

---

## 错误代码

---

0xC0000005 错误属于访问异常

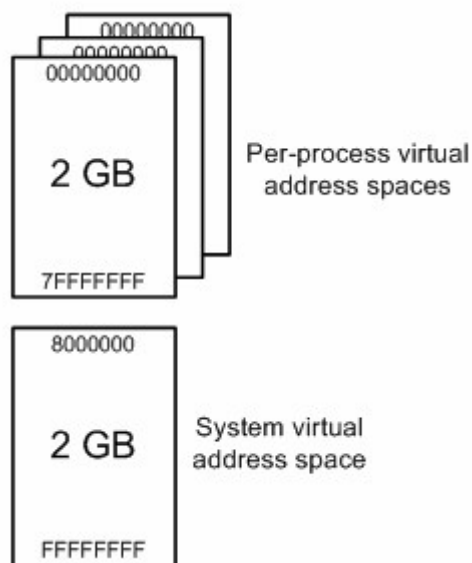
0xC00000FD 错误属于栈溢出错误

## 进程空间

---

在32bit-x86环境下，每个进程理论有4G的内存空间

0x00000000 - 0x7fffffff 为用户态， 0x80000000 - 0xffffffff 为内核态



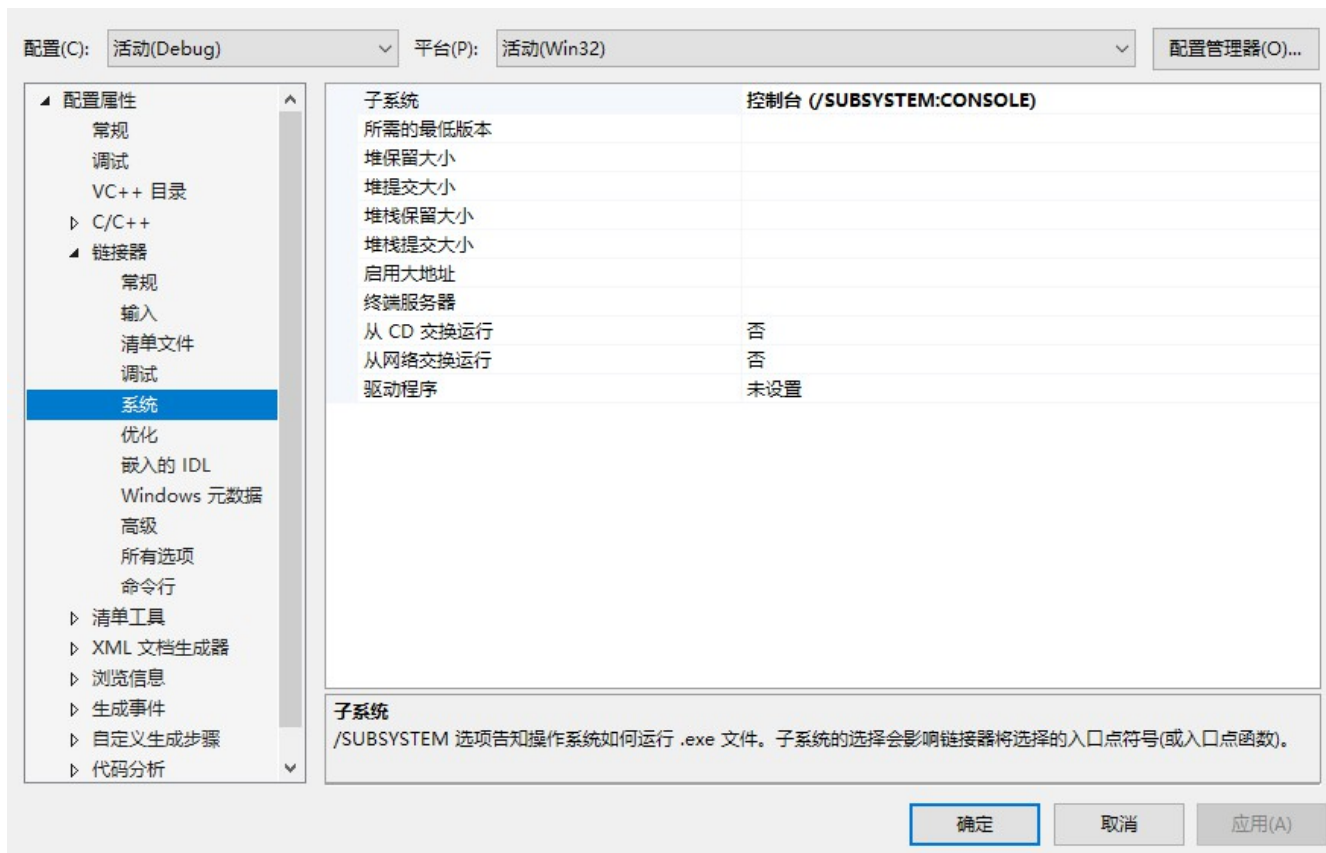
内核态又被称为：**核心态**后者**特权态**

x86体系中具有四种特权等级，特权级别最高的是**ring 0**，被视作内核态；级别最低的是**ring 3**，常被看作用户态；

rings 1 and 2 则很少被使用

## 提交大小与保留大小

---



- 提交大小
  - 目前可以使用大小
- 保留大小
  - 还可以申请的大小

打个比方，马云出门买东西，身上只带了200块，这200块就是提交大小。而马云身价几百个亿，这几百个亿就是保留大小

## 递归

递归函数 `int sum(int n)` 求累加

```
int sum(int)
{
    if(n == 1)
        return 1;
    return sum(n - 1) + n;
}
```

// 调用过程

sum(5)	15
sum(4) + 5	10 + 5 = 15
sum(3) + 4	6 + 4 = 10
sum(2) + 3	3 + 3 = 6

```

|
sum(1) + 2      1 + 2 = 3
|
sum(1) ----- = 1

```

递归调用在线路上比迭代多一倍，在考虑函数调用的效率问题，递归在效率上明显弱于迭代  
线性问题用迭代，非线性可以考虑递归

## 递归的书写技巧

---

1. 先明确递归的结束条件
2. 处理边界或者特殊情况
3. 处理递归主体