偶尔e网事

Work in Cocos2d-x Team.

- 目录视图
- 摘要视图
- 订阅

【C++基础之三】函数中局部变量的返回

标签： C++局部变量指针返回
2013-09-10 14:45 8911人阅读 评论(3) 收藏 举报
分类：
c++ (23)
作者同类文章X

一般说来，函数中是可以进行局部变量的返回的，不然岂不是全部要用全局变量，如果使用了全局变量，那还有必要进行返回吗？那函数就没有它存在的意义了！但是要注意了，这里所谓的局部变量的返回很有内涵，什么样的值才可以进行返回而不出错？

其实，只要遵守一句话即可：函数不能返回指向栈内存的指针！

为什么？因为返回的都是值拷贝！

我们知道，局部变量的作用域是函数内部，函数一旦执行结束，栈上的局部变量会进行销毁，内存得到释放。因此，此时函数返回的是该局部变量的值拷贝，这是没有问题的。但是如果返回的是局部变量的地址，那么返回的只是该局部变量指针的拷贝，而随着函数运行结束，该拷贝指针所指向的栈内存已经被释放，那么指向一个未知区域就会导致调用的错误。

那如果返回的指针指向的是堆内存，又会怎么样？

这样的使用是没有问题的，在函数内new空间，在函数外delete空间。但是这样并不是一种好的编程风格，尽量在同一个作用域内进行new和delete操作，否则还要调用者手动进行内存的释放，试问这样的接口是不是很烂。如果确实需要这样做，那就传指针进去吧！

好吧，通过几个典型的例子看一下，返回局部变量要注意的地方。

1.正确。最normal的情况。

```cpp
int returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<returnValue();
    return 0;
}

char returnValue()
{
    int value=3;
    return value;
}
```

2.错误。最normal错误。虽然value被释放，但是它的值不一定会被清除，所以有时候你这么用看起来结果好像也是对的，但是隐患无穷。

```cpp
int* returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<*(returnValue());
    return 0;
}

int* returnValue()
{
    int value=3;
    return &value;
}
```

3.正确。不用奇怪，“HelloJacky”是一个字符串常量，储存在只读数据段，return str只是返回了该字符串在只读数据段所在的首地址，当函数退出后，该字符串所在的内存不会被回收，所以是正常的。

```cpp
char* returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<returnValue();
    return 0;
}

char* returnValue()
{
    char* str="HelloJacky";
    return str;
}
```

4.错误。这一回“HelloJacky”是栈内的局部变量，函数退出时内存被释放，因此返回栈内局部变量的地址是错误的。

```cpp
char* returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<returnValue();
    return 0;
}

char* returnValue()
{
    char str[]="HelloJacky";
    return str;
}
```

5.正确。如果你非要返回一个局部变量的地址，那么加上static吧。

```cpp
char* returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<returnValue();
    return 0;
}

char* returnValue()
{
    static char str[]="HelloJacky";
    return str;
}
```

6.错误，一样的，数组也不能作为函数的返回值，因为数组名其实是局部变量的首地址。

```cpp
```

```
int* returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<*(returnValue());
    return 0;
}

int* returnValue()
{
    int value[3]={1,2,3};
    return value;
}
```

7.正确。加上static修饰符吧，那数组也可以返回了。

```
[cpp]

int* returnValue();

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<*(returnValue());
    return 0;
}

int* returnValue()
{
    static int value[3]={1,2,3};
    return value;
}
```

8.正确。函数内申请空间，调用后释放空间，只是这样做的坏处就如上面所说接口不灵活。

```
[cpp]

char* newMemory(int size);

int _tmain(int argc, _TCHAR* argv[])
{
    char* p=newMemory(2);
    if(p!=NULL)
    {
        *p='a';
    }
    std::cout<<*p;
    delete [] p;
    return 0;
}

char* newMemory(int size)
{
    char* p=NULL;
    p=new char[size];
    return p;
}
```


- 上一篇【C++基础之二】常量指针和指针常量
- 下一篇【C++基础之四】深拷贝和浅拷贝

相关文章推荐

- • CString 操作指南
- • java 自学笔记 基础篇