

二级C 函数：内联函数(inline) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/133/2021_2022__E4_BA_8C_E7_BA_A7C___E5_c97_133528.htm 在c中，为了解决一些频繁

调用的小函数大量消耗栈空间或者是叫栈内存的问题，特别的引入了inline修饰符，表示为内联函数。可能说到这里，很多人还不明白什么是栈空间，其实栈空间就是指放置程序的局部数据也就是函数内数据的内存空间，在系统下，栈空间是有限的，如果频繁大量的使用就会造成因栈空间不足所造成的程序出错的问题，函数的死循环递归调用的最终结果就是导致栈内存空间枯竭。下面我们来看一个例子：

```
#include <iostream>
#include <string>
using namespace std;
inline string dbtest(int a) //函数原形声明为inline即:内联函数
void main() {
for (int i=1;i { cout } cin.get(). }
string dbtest(int a)//这里不用再次inline,当然加上inline也是不会出错的
{ return (a%2>0)?"奇":"偶". }
}
上面的例子就是标准的内联函数的用法，使用inline修饰带来的好处我们表面看不出来，其实在内部的工作就是在每个for循环的内部所有调用dbtest(i)的地方都换成了(i%2>0)?"奇":"偶"这样就避免了频繁调用函数对栈内存重复开辟所带来的消耗。说到这里很多人可能会问，既然inline这么好，还不如把所谓的函数都声明成inline，嗯，这个问题是要注意的，inline的使用是有所限制的，inline只适合函数体内代码简单的函数使用，不能包含复杂的结构控制语句例如while switch，并且不能内联函数本身不能是直接递归函数(自己内部还调用自己的函数)。说到这里我们不得不说一下在c语言中广泛被使用的#define语句，是的define的确也可以做到inline的这些
```

工作，但是define是会产生副作用的，尤其是不同类型参数所导致的错误，由此可见inline有更强的约束性和能够让编译器检查出更多错误的特性，在c中是不推荐使用define的。关于内联函数的更多例子我就不一一举出了，灵活的使用也多靠学习者本身，我只在此抛砖引玉，让大家尽可能多的学习到c中的一些新的先进的特性知识点。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com