

C语言中的面向对象(3) - 类模拟的性能分析 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E4_B8_AD_c97_135014.htm 类模拟中使用了大量的函数指针，结构体等等，有必须对此进行性能分析，以便观察这样的结构对程序的整体性能有什么程度的影响。

1. 函数调用的开销

```
#define COUNTER XX
void testfunc(){ int i,k=0.
for(i=0.i}
```

在测试程序里面，我们使用的是一个测试函数，函数体内部可以通过改变YY的值来改变函数的耗时。测试对比是循环调用XX次函数，和循环XX次函数内部的YY循环。结果发现，在YY足够小，X足够大的情况下，函数调用耗时成为了主要原因。所以当“简单”功能需要“反复”调用的时候，将它编写为函数将会对性能有影响。这个时候可以使用宏，或者inline关键字。但是，实际上我设置XX = 10000000（1千万）的时候，才出现ms级别的耗时，对于非实时操作（UI等等），即使是很慢的cpu（嵌入式10M级别的），也只会出现在XX = 10万的时候出现短暂的函数调用耗时，所以实际上这个是可以忽略的。

2. 普通函数调用和函数指针调用的开销

```
void (*tf)().tf=testfunc.
```

测试程序修改为一个使用函数调用，一个使用函数指针调用。测试发现对时间基本没有什么影响。（在第一次编写的时候，发现在函数调用出现耗时的情况下（XX = 1亿），函数指针的调用要慢（release版本），调用耗时350：500。后来才发现这个影响是由于将变量申请为全局的原因，全局变量的访问要比局部变量慢很多）。

3. 函数指针和指针结构访问的开销

```
struct a { void (*tf)().}
```

测试程序修改为使用结构的函数指针，测试发现对时间基本没

有什么影响。其实使用结构并不会产生影响，因为结构的访问是固定偏移量的。所以结构变量的访问和普通变量的访问对于机器码来说是一样的。测试结论：使用类模拟的办法对性能不会产生太大的影响。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