

C语言高效编程的四大绝招 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/261/2021_2022_C_E8_AF_AD_E8_A8_80_E9_AB_98_c67_261316.htm

编写高效简洁的C语言代码，是许多软件工程师追求的目标。本文就是针对编程工作中的一些体会和经验做相关的阐述。第一招：以空间换时间 计算机程序中最大的矛盾是空间和时间的矛盾，那么，从这个角度出发逆向思维来考虑程序的效率问题，我们就有了解决问题的第1招--以空间换时间。比如说字符串的赋值：方法A：通常的办法

```
_NOBR> _CODE>#define LEN 32 char string1 [LEN]. memset (string1,0,LEN). strcpy (string1,"This is a example!!" ) _CODE> _NOBR>
```

方法B：_NOBR>_CODE>const char string2[LEN] ="This is a example!". char * cp. cp = string2 _CODE> _NOBR>

使用的时候可以直接用指针来操作。从上面的例子可以看出，A和B的效率是不能比的。在同样的存储空间下，B直接使用指针就可以操作了，而A需要调用两个字符函数才能完成。B的缺点在于灵活性没有A好。在需要频繁更改一个字符串内容的时候，A具有更好的灵活性；如果采用方法B，则需要预存许多字符串，虽然占用了大量的内存，但是获得了程序执行的高效率。如果系统的实时性要求很高，内存还有一些，那我推荐你使用该招数。该招数的变招--使用宏函数而不是函数。举例如下：方法C：

```
_NOBR> _CODE>#define bwMCDR2_ADDRESS 4 #define bsMCDR2_ADDRESS 17 int BIT_MASK(int __bf) { return ((1U } void SET_BITS(int __dst, int __bf, int __val) { __dst = ((__dst) amp. (BIT_MASK(__bf)))) } SET_BITS(MCDR2,
```

MCDR2_ADDRESS, RegisterNumber)._CODE> _NOBR>

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