

新手学堂：嵌入式系统编程中的代码优化 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/271/2021_2022__E6_96_B0_E6_89_8B_E5_AD_A6_E5_c103_271760.htm 在当前计算机技术高度发展的今天，嵌入式系统已经广泛地应用于工业控制、信息家电、办公自动化、移动通信、仪器仪表、医疗电子以及国防等领域。随着国内外各种嵌入式产品的进一步开发和推广，嵌入式技术越来越和人们的生活紧密结合。嵌入式系统一般指非PC系统，通常完成一种或多种特定的计算机功能。

它是以应用为中心，软硬件可裁减的，适应应用系统对功能，可靠性，成本，体积，功耗等综合性要求的专用计算机系统。简单的说类似于PC中的BIOS的工作方式，具有软件代码小、高度自动化、响应速度快等特点。特别适合于要求实时和多任务的应用体系。嵌入式实时系统是目前蓬勃发展的行业之一。但是，实时嵌入式系统的特点使得其软件受时间和空间的严格限制，加上运行环境复杂，使得嵌入式系统软件的开发变得异常困难。为了设计一个满足功能、性能和时间要求的安全可靠的高性能嵌入式系统，编程语言的选择十分重要。

1、嵌入式系统中编程语言的选择

因为汇编语言编写的代码难懂,从而不好维护和难于调试，且只能针对特定的体系结构和处理器移植性差，所以既不宜在复杂系统中使用，又不便于实现软件重用；而高级语言具有良好的通用性和丰富的软件支持，可移植性好、易于维护，因此高级语言编程具有许多优势。随着嵌入式系统应用范围的不断扩大和嵌入式实时操作系统RTOS(Real Time Operating System)的广泛使用，高级语言编程已是嵌入式系统设计的必然趋势。但是 不排

除一些软件模块仍用汇编语言来写，这可以使程序更加有效。虽然C/C++编译器对代码进行了优化，但是适当的使用内联汇编指令可以有效的提高整个系统运行的效率。目前，在嵌入式系统开发过程中使用的语言种类很多，但仅有少数几种语言得到了比较广泛的应用。其中C和C++是应用最广泛的。C++在支持现代软件工程、OOP(Object Oriented Programming, 面向对象的程序设计)、结构化等方面对C进行了卓有成效的改进，但在程序代码容量、执行速度、程序复杂程度等方面比C语言程序性能差一些。由于C语言既有低级语言的直接控制硬件的能力，又有高级语言的灵活性，是目前在嵌入式系统中应用最广泛的编程语言。随着网络技术和嵌入式技术的不断发展，Java的应用也得到广泛应用。

2、实时程序设计中代码的优化

在嵌入式的系统开发中，出于对低价产品的需求，硬件的设计者需要提供刚好足够的存储器和完成工作的处理能力。所以在嵌入式软件设计的最后一个阶段则变成了对代码的优化。代码优化的目标是体积小和速度快,可以从算法、数据和指令流三方面来考虑。算法优化大多数情况下，速度同内存（或者是性能，比如说压缩性能）是不可兼得的。目前程序加速的常用算法一个大方面就是利用查表来避免计算（比如在jpg有huffman码表，在YUV到RGB变换也有变换表）这样原来的复杂计算现在仅仅查表就可以了，虽然浪费了内存，不过速度显著提升。此外在编写程序时还要注意提高效率，例如：Switch语句中根据发生频率来进行case排序

switch语句是一个普通的编程技术，编译器会产生if-else-if的嵌套代码，并按照顺序进行比较，发现匹配时，就跳转到满足条件的语句执行。使用时需要注意。每一个由机器语言实现

的测试和跳转仅仅是为了决定下一步要做什么，就把宝贵的处理器时间耗尽。为了提高速度，设法根据具体的情况按照它们发生的相对频率排序。换句话说，把最可能发生的情况放在第一位，最不可能的情况放在最后。将大的switch语句转为嵌套switch语句 当switch语句中的case标号很多时，为了减少比较的次数，明智的做法是把大switch语句转为嵌套switch语句。把发生频率高的case 标号放在一个switch语句中，并且是嵌套switch语句的最外层，发生相对频率相对低的case标号放在另一个switch语句中。如果switch中每一种情况下都有很多的工作要做，那么把整个switch语句用一个指向函数指针的表来替换会更加有效。用指针代替数组 在许多种情况下，可以用指针运算代替数组索引，这样做常常能产生又快又短的代码。与数组索引相比，指针一般能使代码速度更快，占用空间更少。使用多维数组时差异更明显。下面的代码作用是相同的，但是效率不一样。 数组索引 指针运算

```
For(..){ p=array A=array[r ]. for(..){ a=*(p ). ..... } } 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