

C语言中volatile关键字的学习 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_C_E8_AF_AD_E8_A8_80_E4_B8_AD_c97_646598.htm 导读:volatile关键字是一种类型修饰符，用它声明的类型变量表示可以被某些编译器未知的因素更改。百考试题免费提供大量实用资料，本次辅导将全部采用视频授课的形式呈现给广大学员,考生可以随时报名参加学习，不用东奔西跑搜寻资料，备考费用更加低廉！#0000ff>欲报从速！用volatile关键字声明的变量i每一次被访问时，执行部件都会从i相应的内存单元中取出i的值。没有用volatile关键字声明的变量i在被访问的时候可能直接从cpu的寄存器中取值（因为之前i被访问过，也就是说之前就内存中取出i的值保存到某个寄存器中），之所以直接从寄存器中取值，而不去内存中取值，是因为编译器优化代码的结果（访问cpu寄存器比访问ram快的多）。以上两种情况的区别在于被编译成汇编代码之后，两者是不一样的。之所以这样做是因为变量i可能会经常变化，保证对特殊地址的稳定访问。volatile关键字是一种类型修饰符，用它声明的类型变量表示可以被某些编译器未知的因素更改，比如：操作系统、硬件或者其它线程等。遇到这个关键字声明的变量，编译器对访问该变量的代码就不再进行优化，从而可以提供对特殊地址的稳定访问。使用该关键字的例子如下：`int volatile nVint`. 当要求使用volatile 声明的变量的值的时候，系统总是重新从它所在的内存读取数据，即使它前面的指令刚刚从该处读取过数据。而且读取的数据立刻被保存。例如：`volatile int i=10. int a = i. ... //其他代码，并未明确告诉编译器，对i进`

行过操作 `int b = i`。 `volatile` 指出 `i` 是随时可能发生变化的，每次使用它的时候必须从 `i` 的地址中读取，因而编译器生成的汇编代码会重新从 `i` 的地址读取数据放在 `b` 中。而优化做法是，由于编译器发现两次从 `i` 读数据的代码之间的代码没有对 `i` 进行过操作，它会自动把上次读的数据放在 `b` 中。而不是重新从 `i` 里面读。这样以来，如果 `i` 是一个寄存器变量或者表示一个端口数据就容易出错，所以说 `volatile` 可以保证对特殊地址的稳定访问。 注意，在 `vc6` 中，一般调试模式没有进行代码优化，所以这个关键字的作用看不出来。下面通过插入汇编代码，测试有无 `volatile` 关键字，对程序最终代码的影响：首先，用 `classwizard` 建一个 `win32 console` 工程，插入一 `voltest.cpp` 文件，输入下面的 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