

第5章 C++内存模型和原子类型操作

本章主要内容

- C++11内存模型详解
- 标准库提供的原子类型
- 使用各种原子类型
- 原子操作实现线程同步功能

C++11标准中，有一个十分重要特性，常被程序员们所忽略。它不是一个新语法特性，也不是新工具，它就是多线程(感知)内存模型。内存模型没有明确的定义基本部件应该如何工作的话，之前介绍的那些工具就无法正常工作。那为什么大多数程序员都没有注意到它呢？当你使用互斥量保护你的数据和条件变量，或者是“期望”上的信号事件时，对于互斥量为什么能起到这样作用，大多数人不会去关心。只有当你试图去“接触硬件”，你才能详尽的了解到内存模型是如何起作用的。

C++是一个系统级别的编程语言，标准委员会的目标之一就是不需要比 C++ 还要底层的高级语言。C++ 应该向程序员提供足够的灵活性，无障碍的去做他们想要做的事情；当需要的时候，可以让他们“接触硬件”。原子类型和原子操作就允许他们“接触硬件”，并提供底层级别的同步操作，通常会常规指令数缩减到1~2个CPU指令。

本章，我们将讨论内存模型的基本知识，而后再了解一下原子类型和操作，最后了解与原子类型操作相关的各种同步。这个过程会比较复杂：除非你已经打算使用原子操作(比如，第7章的无锁数据结构)同步你的代码，否则，就没有必要了解过多的细节。

让我们先轻松愉快的来看一下有关内存模型的基本知识。