链滴

# 面试题之 golang 语言篇

作者：zhengliwei

原文链接：https://ld246.com/article/1636861879229

来源网站：链滴

<h2 id="golang协程和线程的区别">golang 协程和线程的区别</h2>
<ul>
<li>线程是操作系统负责调度的，调度时需要切换到内核态；golang 协程也称用户态线程，是由 golang 运行时负责调度的，完全在用户态进行调度。由于没有切换到内核态的开销，golang 协程的调度比线程调度快很多。</li>
<li>相比线程，golang 协程占用内存空间很小，再加上调度快，golang 可以轻松地支持数百万的协并发。</li>
</ul>
<h2 id="数组和切片的区别与联系">数组和切片的区别与联系</h2>
<ul>
<li>数组和固定长度的，在创建时就要指定长度，且后续长度不可变，准确地说，数组的长度也是数类型的一部分，比如[1]string 和[2]string 是不同的两种类型；而切片长度可变，在不断往切片里添元素后切片会进行扩容。</li>
<li>切片可以看做是对数组的一层封装，切片中维护了指向底层数组的指针，以及在底层数组中的开位置和结束位置，因此，在使用切片时需要注意：修改切片的元素会影响底层数组，尤其在多个切片用同一个底层数组时，会相互影响。</li>
<li>需要注意的时，切片永远不会替换底层数组，在调用 append()函数给切片增加元素时，如果需做扩容，会新创建一个底层数组，同时也会新建一个切片，作为 append()函数的返回值，原来的切与原来的底层数组的对应关系不会改变。</li>
</ul>
<h2 id="简单介绍下golang协程调度模型">简单介绍下 golang 协程调度模型</h2>
<ul>
<li>golang 运行时实现的协程调度模型被称为 GMP 模型。G 指 goroutine，即 golang 协程；M 指 mathine，即物理线程；P 指 processor，即调度器，由 golang 运行时实现。</li>
<li>processor 会维护一个待运行的协程队列，并与一个物理线程关联，processor 会从待运行的协队列里取出一个协程与物理线程进行绑定，开始执行协程代码，当协程遇到阻塞事件时，processor 把协程与物理线程解绑，并取出下一个协程绑定到物理线程中开始执行，这个切换动作完全是在用户进行的，并不涉及内核态的上下文切换，因此很快。</li>
</ul>