见javaguide

[https://javaguide.cn/cs-basics/operating-system/operating-system-basic-questions-01.html#%E4%BB%80%E4%B9%88%E6%98%AF%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F](https://javaguide.cn/cs-basics/operating-system/operating-system-basic-questions-01.html" \l "%E4%BB%80%E4%B9%88%E6%98%AF%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)

# CPU上下文切换

<https://zhuanlan.zhihu.com/p/52845869>

CPU 寄存器和程序计数器就是 CPU 上下文

**什么是 CPU 上下文切换**

就是先把前一个任务的 CPU 上下文（也就是 CPU 寄存器和程序计数器）保存起来，然后加载新任务的上下文到这些寄存器和程序计数器，最后再跳转到程序计数器所指的新位置，运行新任务。

而这些保存下来的上下文，会存储在系统内核中，并在任务重新调度执行时再次加载进来。这样就能保证任务原来的状态不受影响，让任务看起来还是连续运行。

## 怎么减少并发中的上下文切换？

1. 无锁并发编程。多线程竞争锁时，会引起上下文切换，所以多线程处理数据时，可以用一  
些办法来避免使用锁，如将数据的ID按照Hash算法取模分段，不同的线程处理不同段的数据。  
2. CAS算法。Java的Atomic包使用CAS算法来更新数据，而不需要加锁。  
3. 使用最少线程。避免创建不需要的线程，比如任务很少，但是创建了很多线程来处理，这  
样会造成大量线程都处于等待状态。  
4. 协程：在单线程里实现多任务的调度，并在单线程里维持多个任务间的切换。

# 进程间的通信方式有哪些？

见javaguide

# 进程与线程

## 进程与线程之间的区别？

https://www.zhihu.com/question/25532384/answer/81152571

## 线程之间共享了哪些进程资源？

https://cloud.tencent.com/developer/article/1768025