

题目 01- 请你说一说什么是线程和进程？

- 区别

A: 进程是计算机运行的程序，是运行程序的基本单位。进程拥有自己独立的内存空间。

线程是进程最基本的执行单元，一个进程可以有多个线程。

- 关系

A: 进程的内存空间是独立的，内存数据互不干扰。线程的堆和方法区是共享的，但是程序计数器和栈空间是独立的，线程的资源开销比进程小很多。

- 线程的上下文切换是什么？

A: 现代 cpu 的调度方式是采用时间片的抢占式调度。每个线程执行完分配的时间片后，将当前线程执行的指令位置记录到程序计数器中，后续当前线程抢占的 cpu 后从程序计数器中恢复到上次执行的指令位置继续执行的过程就是线程上下文切换。

- 线程的并发与并行有啥区别？

A: 线程的并发执行是指同一时间内能够处理多个任务，但是单位时间内不一定是同时执行的。线程的并行是指同一时间内同时处理多个任务，单位时间内一定是同时执行的。可以说，并发是一种能力，并行是一种手段。当并发的代码运行在多核 cpu 中可能产生并行执行。

题目 02- 使用了多线程会带来什么问题呢？

- 能不能详细说说线程安全问题？

A: 线程安全问题是指多个线程在对同一个共享资源操作时，与单线程的执行预期结果不一致问题。

- 原子性、有序性和可见性能不能深入的谈一下。

A: 原子性：一个或多个操作要么全部执行，要么都不执行。

有序性：cpu 为了提高执行的效率，可能对执行的指令重排序。有序性是指在指令重排的情况下，仍能够保证正确的执行结果，就像 as-if-serial 一样。

可见性：cpu 调度线程执行指令时，执行指令很快，但是从内存中读取数据很慢。所以为了提高数据读取的速度，cpu 设置了多级高速缓存区。cpu 执行指令时，只能通过线程的本地内存(高速缓存区)读取数据，写入数据时，必须通过本地内存写入到直接内存。可见性就是指在多线程的情况下，一个线程对共享变量的写，能够立马刷到本地内存，线程对共享变量的读，能够从本地内存中获取到最新的值。

题目 03- 什么是死锁？如何排查死锁？

排查过程最好详细说明，最少说一种排查方案，越多越好。

A: 死锁是指两个或多个进程/线程在互相等待对方所占有的资源而无法继续执行的情况。

每个进程/线程都在等待其他进程释放资源，但是这种循环等待导致所有进程/线程都无法继续执行，进而造成系统锁死。

排查手段：

- 1.通过 jstack、visualVm、arthas 可以分析排查 jvm 内部多线程死锁的问题。
- 2.在 java 中使用 synchronized 和 ReentrantLock 等锁机制时,可以通过调试代码和观察日志来查看锁对的使用情况，查看共享资源的占用情况。