

11 。流程 process 和线程 thread 之间的区别是什么？

流程是一个程序的执行，而线程是流程中的一个单一的执行顺序。一个流程可以包含多个线程。一个线程有时被称为轻量级流程。

12 。请说明创建线程的不同方式。哪一个你会喜欢，为什么？

创建一个线程可以使用三种方式：

一个类可以继承 Thread 类。

一个类可以实现 Runnable 接口。

一个应用程序可以使用的 Executor 框架，以创建一个线程池。

传统情况下 Runnable 接口是首选，因为它并不需要一个对象来继承 Thread 类。如果您的应用程序设计需要多重继承，只有接口可以帮助你。还有，线程池是非常有效的，并且可以被实现和非常容易使用。在并发多核编程中，使用 Executor 框架或 ForkJoin 是首选。

13 。请从一个高阶水平讲解一个线程状态。

一个线程在执行时可以有以下状态之一：

Runnable 可运行的：一个线程准备好运行，但不一定立即开始运行。

Running 运行：处理器正在积极执行的线程代码。

Waiting 等待：一个线程处于阻塞状态，等待一些外部处理完成。

Sleeping 睡：线程都被迫睡觉。

阻塞 I / O ：等待 I / O 操作完成。

阻塞同步：等待获取锁。

Dead 死者：该线程已经完成其执行。

14 。一个 `synchronized` 方法和 `synchronized` 块之间的区别是什么？

在 Java 编程中，每个对象都有一个锁。一个线程可以使用 `synchronized` 关键字获得锁的对象。 `synchronized` 关键字可以在方法级（粗粒度锁）或代码块级别（细粒度锁）被应用。

15 。线程同步在监视器的内部如何发生？你可以使用什么水平的同步？

JVM 通过锁与监视器配合使用。监视器基本上是监视的同步码序列，并确保某个时间只有一个线程执行同步的代码块。每个监视器都与一个对象引用相关联。线程只有获得锁以后才可以执行该代码。

16 。什么是死锁？

当两个流程在继续操作之前都在等待对方完成，其结果是，这两个流程进入等待不休的循环。

17 。你如何确保 N 个线程可以访问 N 个资源而没有死锁？

一个非常简单的方法，同时使用 N 个线程，以避免死锁是强加一个顺序上的锁，并迫使每个线程遵循这一顺序。因此，如果所有线程对互斥体 `mutex` 锁定和解锁都是以相同的顺序，那就没有死锁可能发生。