面试真题

【美团】实现多线程编程的方式有哪些？

**实现多线程编程的方式主要有 三种：**

1. **继承Thread类**：创建一个继承Thread类的子类，重写run()方法。
2. **实现Runnable接口**：实现Runnable接口，并将其传递给Thread对象。
3. **使用Executor框架**：通过Executor接口及其实现（如ThreadPoolExecutor）管理线程池，推荐使用此方法。

【美团】创建线程池有哪些方式？

1）使用 Executors 工厂类，例如Executors.newFixedThreadPool(10);

2）使用 ThreadPoolExecutor 直接创建线程池

3）通过 ForkJoinPool 创建并行任务线程池。

【美团】线程池的核心参数？执行流程？

* corePoolSize：核心线程数，即线程池中始终保持的线程数量。
* maximumPoolSize：最大线程数，即线程池中允许的最大线程数量。
* keepAliveTime：线程空闲时间，超过这个时间的非核心线程会被销毁。
* workQueue：任务队列，存放待执行的任务。
* threadFactory：线程工厂，用于创建新线程。
* rejectedExecutionHandler：任务拒绝处理器，当任务无法执行时的处理策略。

【美团】线程有哪些状态？

**Java线程的状态有 五种，分别是：**

* **NEW**：线程创建后，尚未调用start()方法时的状态。
* **RUNNABLE**：线程处于运行状态，可能正在CPU上执行，也可能在等待CPU资源。
* **BLOCKED**：线程因获取锁而被阻塞，等待其他线程释放锁。
* **WAITING**：线程在等待某些条件满足时进入等待状态，调用wait()、join()等方法时会进入此状态。
* **TERMINATED**：线程执行完毕或被中断，进入终止状态。

【美团】现在有线程A和B如何实现A运行完以后可以再运行B？

**实现A运行完后再运行B的方式：**

1. **使用Thread.join()方法**：在线程A执行完毕后，调用A.join()使得B等待A结束后再执行。
2. **使用CountDownLatch**：通过CountDownLatch来控制B的执行，直到A完成计数后再启动B。
3. **使用ExecutorService的顺序执行**：通过线程池确保A任务执行完后，B任务才会执行。

【美团】如果服务器出现了线程泄露，你应该怎么怎么去排查解决？

1. **使用线程Dump分析线程堆栈**：
   * 通过jstack命令或jvisualvm等工具查看线程的堆栈信息。如果发现有大量线程处于WAITING、BLOCKED或TIMED\_WAITING状态且长时间未变，则可能是线程泄露。
   * 分析输出内容，查看哪些线程在等待锁，是否存在线程无法结束的情况。
2. **查看线程池配置**：
   * 线程池大小配置不当会导致任务积压或线程无法回收。通过检查ExecutorService等线程池配置，确保最大线程数、核心线程数等参数适合当前的负载。
   * 确保线程池在任务执行完后能正确回收线程，并且不会被长时间占用。
3. **分析业务逻辑**：
   * 查找可能导致线程无法结束的代码，例如：
     + 网络或数据库连接没有及时关闭。
     + 锁未释放（死锁）导致线程阻塞。
   * 确保所有finally块中能正确关闭资源，如InputStream、OutputStream、Connection等。
4. **使用性能监控工具**：
   * 使用VisualVM、JProfiler、YourKit等工具监控JVM线程的情况，查找哪些线程数量异常增加，或者存在大量长时间运行的线程。
   * 监控工具可以提供线程的实时视图，帮助定位问题线程和线程泄露的根本原因。
5. **检查线程池使用情况**：
   * 线程池泄露常见问题是：任务未完成，线程无法回收。
   * 确保线程池在任务完成后能够正确清理。通过调用shutdown()或shutdownNow()来确保线程池停止工作，并释放资源。
   * 还可以配置拒绝策略来避免无限制地提交任务（如AbortPolicy、CallerRunsPolicy等）。

**延申问答**

* **如何优化线程池？**
  + 使用合适的ThreadPoolExecutor配置，确保线程池的大小与实际工作负载匹配。
  + 设置合理的keepAliveTime，确保空闲线程能及时回收。
  + 使用BlockingQueue控制任务的提交，防止任务积压。

面向简历

Java并发容器和锁

ConcurrentHashMap

CAS

AQS

知识框架

Java并发 线程基础与生命周期

线程的生命周期在 Java 中是如何定义的？（中等）

Java 中如何创建多线程？（简单）

什么是 Java 中的线程同步？（中等）

Java 中线程之间如何进行通信？（中等）

Java 中线程安全是什么意思？（中等）

Java并发 线程池与执行框架

Java 创建线程池有哪些方式？

1）使用 Executors 工厂类，例如Executors.newFixedThreadPool(10);

2）使用 ThreadPoolExecutor 直接创建线程池

3）通过 ForkJoinPool 创建并行任务线程池。

你了解 Java 线程池的原理吗？（中等）

Java 线程池有哪些拒绝策略？（中等）

Java 并发库中提供了哪些线程池实现？它们有什么区别？（中等）

Java 线程池核心线程数在运行过程中能修改吗？如何修改？（困难）

Java 线程池中 shutdown 与 shutdownNow 的区别是什么？（中等）

Java 线程池内部任务出异常后，如何知道是哪个线程出了异常？（中等）

Java并发 线程池的高级应用

Java 中的 DelayQueue 和 ScheduledThreadPool 有什么区别？（中等）

什么是 Java 的 Timer？（中等）

你了解时间轮（Time Wheel）吗？有哪些应用场景？（困难）

如何合理地设置 Java 线程池的线程数？（中等）

Java并发 并发工具与同步原语

你使用过哪些 Java 并发工具类？（中等）

什么是 Java 的 Semaphore？（中等）

Java并发 协程与高级并发概念

什么是协程？Java 支持协程吗？（中等）