

并发编程的Java抽象 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E5\\_B9\\_B6\\_E5\\_8F\\_91\\_E7\\_BC\\_96\\_E7\\_c104\\_645255.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E5_B9_B6_E5_8F_91_E7_BC_96_E7_c104_645255.htm) 各种程序员都工作在各自的程序抽象维度，如果我们发现解决一件事情比较难，也许是我们面对的抽象级别还不够高，或者引入的间接程度不够，本文以抽象角度来剖析并发编程。

### 一、机器和OS级别抽象

(1) 冯诺伊曼模型 经典的顺序化计算模型，貌似可以保证顺序化一致性，但是没有哪个现代的多处理架构会提供顺序一致性，冯氏模型只是现代多处理器行为的模糊近似。这个计算模型，指令或者命令列表改变内存变量直接契合命令编程泛型，它以显式的算法为中心，这和声明式编程泛型有区别。就并发编程来说，会显著的引入时间概念和状态依赖 所以所谓的函数式编程可以解决其中的部分问题。

(2) 进程和线程 进程抽象运行的程序，是操作系统资源分配的基本单位，是资源cpu，内存，IO的综合抽象。线程是进程控制流的多重分支，它存在于进程里，是操作系统调度的基本单位，线程之间同步或者异步执行，共享进程的内存地址空间。

(3) 并发与并行 并发，英文单词是concurrent，是指逻辑上同时发生，有人做过比喻，要完成吃完三个馒头的任务，一个人可以这个馒头咬一口，那个馒头咬一口，这样交替进行，最后吃完三个馒头，这就是并发，因为在三个馒头上同时发生了吃的行为，如果只是吃完一个接着吃另一个，这就不是并发了，是排队，三个馒头如果分给三个人吃，这样的任务完成形式叫并行，英文单词是parallel. 回到计算机概念，并发应该是单CPU时代或者单核时代的说法，这个时

候CPU要同时完成多任务，只能用时间片轮转，在逻辑上同时发生，但在物理上是串行的。现在大多数计算机都是多核或者多CPU，那么现在的多任务执行方式就是物理上并行的。为了从物理上支持并发编程，CPU提供了相应的特殊指令，比如原子化的读改写，比较并交换。

（4）平台内存模型在可共享内存的多处理器体系结构中，每个处理器都有它自己的缓存，并且周期性的与主存同步，为什么呢？因为处理器通过降低一致性来换取性能，这和CAP原理通过降低一致性来获取伸缩性有点类似，所以大量的数据在CPU的寄存器中被计算，另外CPU和编译器为了性能还会乱序执行，但是CPU会提供存储关卡指令来保证存储的同步，各种平台的内存模型或者同步指令可能不同，所以这里必须介入对内存模型的抽象，JMM就是其中之一。

二、编程模型抽象

（1）基于线程模型 （2）基于Actor模型 （3）基于STM软件事务内存 .....

Java体系是一个基于线程模型的本质编程平台，所以我们主要讨论线程模型。

三、并发单元抽象 大多数并发应用程序都是围绕执行任务进行管理的，任务是抽象，离散的工作单元，所以编写并发程序，首要工作就是提取和分解并行任务。一旦任务被抽象出来，他们就可以交给并发编程平台去执行，同时在任务抽象还有另一个重要抽象，那就是生命周期，一个任务的开始，结束，返回结果，都是生命周期中重要的阶段。那么编程平台必须提供有效安全的管理任务生命周期的API。

四、线程模型 线程模型是Java的本质模型，它无所不在，所以Java开发必须搞清楚底层线程调度细节，不搞清楚当然就会有struts1，struts2的原理搞不清楚的基本灾难（比如在struts2的action中塞入状态，把struts2的action配成单例

)。用线程来抽象并发编程，是比较低级别的抽象，所以难度就大一些，难度级别会根据我们的任务特点有以下几个类别（1）任务非常独立，不共享，这是最理想的情况，编程压力为0。（2）共享数据，压力开始增大，必须引入锁，Volatile变量，问题有活跃度和性能危险。（3）状态依赖，压力再度增大，这时候我们基本上都是求助jdk提供的同步工具。

五、任务执行 任务是一个抽象体，如果被抽象了出来，下一步就是交给编程平台去执行，在Java中，描述任务的一个基本接口是Runnable，可是这个抽象太有限了，它不能返回值和抛受检查异常，所以Jdk5.0有另外一个高级抽象Callable。任务的执行在Jdk中也是一个底级别的Thread，线程有好处，但是大量线程就有大大的坏处，所以如果任务量很多我们并不能就创建大量的线程去服务这些任务，那么Jdk5.0在任务执行上做了抽象，将任务和任务执行隔离在接口背后，这样我们就可以引入比如线程池的技术来优化执行，优化线程的创建。任务是有生命周期的，所以Jdk5.0提供了Future这个对象来描述对象的生命周期，通过这个future可以取到任务的结果甚至取消任务。

六、锁 当然任务之间共享了数据，那么要保证数据的安全，必须提供一个锁机制来协调状态，锁让数据访问原子，但是引入了串行化，降低了并发度，锁是降低程序伸缩性的原罪，锁是引入上下文切换的主要原罪，锁是引入死锁，活锁，优先级倒置的绝对原罪，但是又不能没有锁，在Java中，锁是一个对象，锁提供原子和内存可见性，Volatile变量提供内存可见性不提供原子，原子变量提供可见性和原子，通过原子变量可以构建无锁算法和无锁数据结构，但是这需要高高手才可以办到。 百考试题温

馨提示：本内容来源于网络，仅代表作者个人观点，与本站立场无关，仅供您学习交流使用。其中可能有部分文章经过多次转载而造成文章内容缺失、错误或文章作者不详等问题，请您谅解。如有侵犯您的权利，请联系我们，本站会立即予以处理。 编辑特别推荐: #0000ff>Java读取文件内容再编辑 #0000ff>JS获取单选与多选按钮的值 #0000ff>每一种文件类型所对应的ContentType 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)