第一章 简介

#### 一．进程与线程

程序

1. 概念

程序是一组指令的有序集合。

进程

1. 概念

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动，是系统进行资源分配和调度的一个独立单位。

1. 进程拥有的资源（操作系统分配之）
2. 内存
3. 文件句柄
4. 安全证书
5. …
6. 进程的通信机制（粗粒度）
7. 套接字
8. 信号处理器
9. 共享内存
10. 信号量
11. 文件
12. …

线程

1. 概念

线程是进程的一个实体，是CPU调度和分派的基本单位，它是比进程更小的能独立运行的基本单位。线程自己基本上不拥有系统资源，只拥有一点在运行中必不可少的资源（如程序计数器，一组寄存器，栈），一个线程可以创建和撤销另一个线程。

1. 私有资源
2. 程序计数器
3. 栈
4. 局部变量
5. …
6. 公有资源（进程范围内的资源）
7. 内存句柄
8. 文件句柄
9. …

4. 优点

（1） 使用得当可以有效降低程序的开发和维护成本，同时提升复杂程序的性能以及资源利用率。发挥多处理器的强大能力

（2） 线程可以将大部分的异步工作流转换成简单的串行工作流，因此能更好的模拟人类的工作方式，使得建模更简单。

（3） 降低代码复杂度，使代码更容易编写和维护。

（4） 提升服务器资源利用率和系统吞吐率。

（5） 简化jvm的实现，如垃圾收集器。

5. 风险

（1） 安全性问题，包括丢失更新等。在寄存器或者处理器中会存在缓存，这些被缓存的变量可以被多个线程共享的话就会出现更新丢失问题。

（2） 活跃性问题，例如死锁等。

（3） 性能问题，包括服务时间长，响应不灵敏，吞吐率低，资源消耗高，可伸缩性低。而且，频繁的上下文切换将带来极大的开销，此时CPU将花费极大的时间在线程切换上而不是线程运行。

6. 常用API

（1） Timer需要是线程安全的。

（2） Servlet和JSP需要是线程安全的。

（3） RMI对象应该做好被多个线程同时调用的准备，并且确保自身是安全的。

一些概念

1. 资源利用率

如果程序必须等待另一个程序运行结束才可以运行，那么可能会降低资源利用率。程序在等待时可以同时运行另一个程序那么必然提高利用率。

2. 公平性

不同用户和程序对于计算机上的资源有着同等的使用权。一种高效的运行方式是通过粗粒度的时间分片使这些用户和程序能共享计算机资源，而不是由一个程序从头运行到尾，然后再启动下一个程序。

3. 便利性

4. 串行性

每次只做一件事，做完再做另一件事。

5. 异步性

同时做多件事，在做一件事的时候不用等待它完成就可以去做另一件事。比方说，注册用户的时候，需要发送验证码短信和邮箱验证。

串行是：注册🡪发送短信🡪发送邮箱🡪验证通过

异步是：注册🡪发送短信🡪验证通过

🡪发送邮箱