# JAVA多线程

### 第1章 进程

1.1 进程为什么出现

1.2 进程的组成

1.3 如何竞争资源（调度算法）

1.3.1 FCFS

1.3.2 RR

1.3.3 SPN

1.3.4 SRT

1.3.5 HRRN

1.3.6 FB

1.4 进程状态图

1.4.1 三态图

1.4.2 五态图

1.4.3 七态图

1.5 进程关系

1.5.1 父子关系

1.5.2 僵尸进程

1.5.3 孤儿进程

1.6 执行模式

1.7 进程间通讯

1.7.1 管道（Pipe）

1.7.2 流管道（Flow Pipe）

1.7.3 有名管道（Named Pipe）

1.7.4 信号量（Semophore）

1.7.5 信号（Signal）

1.7.6 消息队列（Message Queue）

1.7.7 共享内存（Shared Memory）

1.7.8 套接字（Socket）

### 第2章 概念

2.1 同步 & 异步

2.2 串行 & 并行 & 并发

2.3 阻塞 & 非阻塞

### 第3章 线程

3.1 什么是线程

3.2 进程与线程区别

3.3 线程状态图

3.4 Java 线程实现

3.4.1 基础 Thread 类

3.4.2 实现 Runable 接口

3.5 线程各方法介绍

3.5.1 sleep()

3.5.2 yield()

3.5.3 join()

…

3.6 线程优先级

3.7

### 第4章 Java 如何支持多线程

4.1 多线程带来的问题

4.1.1 线程安全

4.1.1.1 原子性

4.1.1.2 可见性

4.1.1.3 有序性

4.2 synchronized

4.3 volatile

4.3 happen before 原则

4.4 线程通讯

4.4.1 等待/通知机制

4.4.2 join() 使用

4.4.3 ThreadLocal 使用

### 第5章 锁

5.1 锁的含义

5.2 读写锁

5.3 死锁

5.4 锁消除

5.5 锁粗化

5.6 偏向锁

5.7 适应性锁

5.8 AQS

### 第6章 原子类

6.1 基本类型

6.2 数组类型

6.3 引用类型

6.4 更新字段

### 第7章 线程池

7.1 线程池的组成

7.2 Java线程池的分类

### 第8章 定时器

8.1 定时器分类

8.2 定时器使用

8.3 定时器案例

### 第9章 并发容器

9.1 Vector

9.2 ConcurrentHashMap

9.3 CocurrentLinkedQueue

9.4 Fork/Join 框架

### 第10章 多线程设计模式

10.1 Immutable Object(不可变对象)模式

10.2 Guarded Suspension(保护性暂挂)模式

10.3 Two-phase Termination(两阶段终止)模式

10.4 Promise(承诺)模式

10.5 Producer-Consumer(生产者消费者)模式

10.6 Active Object(主动对象)模式

10.7 Thread Pool(线程池)模式

10.8 Thread Specific Storage(线程特有存储)模式

10.9 Serial Thread Confinement(串行线程封闭)模式

10.10 Master-Slave(主仆)模式

10.11 Pipeline(流水线)模式

10.12 Half-sync/Half-async(半同步/半异步)模式

10.13 Worker-Thread设计模式

10.14 Event Bus设计模式

10.15 Event Driven设计模式

10.16 Latch设计模式

10.17 Balking设计模式

10.18 Future设计模式

10.19 读写锁分离设计模式

10.20 Single Thread Execution设计模式