# 3.1.进程

#### 程序的运行方式

##### 顺序运行

##### 并发运行（和共享式相辅相成的）

##### 并行运行

# 2.进程特征与控制（5.20）

### 2.1.特征

#### 结构性

#### 动态性

#### 独立性

#### 并发性

### 2.2.进程分类

#### 系统进程

#### 用户进程

### 2.3.进程状态转化

#### 就绪状态

#### 运行状态

#### 阻塞状态

#### 新建状态

#### 终止状态

#### 挂起就绪状态

#### 挂起阻塞状态

### 2.4.进程控制块

#### 创建进程

#### 撤销和终止进程

#### 阻塞与唤醒进程

#### 挂起和激活进程

### 2.5.进程的互斥： 死锁，饥饿

#### 2.5.1与时间有关的错误

#### 2.5.2临界资源与临界区

##### 2.5.2.1软件方法管理临界区

##### 2.5.2.2硬件设施管理临界区

###### 禁止中断法

###### 特殊指令法：TS指令，SWAP指令

### 2.6.进程的同步（涉及并发，正好想要了解网络编程中的多用户并发）

#### 互斥是一种特殊的同步

#### 经典问题

##### 1.生产者-消费者问题（数据丢失，缓冲区溢出问题）

##### 2.读者-写者问题

##### 3.哲学家就餐问题（死锁和饥饿）

##### 4.睡眠理发师问题

### 2.7.进程通信

#### 消息传递通信

#### 共享内存通信

#### 管道通信

### 2.8.进程调度

#### 调度模型

##### 高级，中级，低级

#### 调度算法

##### 先来先服务

##### 短作业优先（有贪心的影子）

##### 最短剩余时间优先

##### 高相应比优先

##### 优先权

##### 时间片轮转

##### 多级反馈队列

#### 多CPU系统中的调度

#### 多核CPU中的调度

### 2.9.死锁

### 2.10.线程

#### 1.在操作系统内核实现的内核级线程

#### 2.在用户空间实现的用户级线程

#### 3.同时支持两种方式的线程

### 2.11.管程

#### 1.特点

##### 模块化，隐蔽化，互斥性

#### 2.可以解决进程同步经典问题的1和3

##### 1.生产者-消费者问题

##### 3.哲学家就餐问题（说明同步问题的重要性）

### 2.12Linux中进程与线程，Android中的进程与线程，Windows中的进程与线程