

## 进程和线程

- 进程是操作系统资源分配的基本单位，而线程是任务调度和执行的基本单位
- 每个进程都有独立的代码和数据空间，程序之间的切换会有较大的开销；线程之间的切换开销小，同一类线程共享代码和数据空间，每个线程有着自己独立的运行栈和程序计数器。
- 在操作系统中能同时运行多个进程（程序）；而在同一个进程（程序）中有多个线程同时执行（通过CPU调度，在每个时间片中只有一个线程执行）

## 进程间的通信方式

- 无名管道（PIPE）
- 命名管道（FIFO）
- 消息队列
- 信号量
- 共享内存

## CFS调度策略

- O(n)调度：在每次进程切换时，内核依次扫描就绪队列上的每一个进程，计算每个进程的优先级，再选择出优先级最高的进程来运行。
- O(1)调度：其基本思想是根据进程的优先级进行调度。进程有两个优先级,一个是静态优先级,一个是动态优先级.静态优先级是用来计算进程运行的时间片长度的,动态优先级是在调度器进行调度时用到的,调度器每次都选取动态优先级最高的进程运行。
- CFS调度：不计算优先级，而是通过计算进程消耗的CPU时间（标准化以后的虚拟CPU时间）来确定调度
  - 绝对公平性：将CPU当做一种资源，调度时总选择消耗资源最少的进程，这就是所谓的完全公平。但不同重要性的进程的优先级会比较高，故而我们希望能按照权重来重新分配CPU资源。
  - 相对公平性：根据不同进程的权重分配时间。运行时间的计算公式为：分配给进程的运行时间 = 调度周期 \* 进程权重 / 所有进程权重之和
  - linux通过引入virtual runtime（vruntime）来完成上述设想。具体我们可以看下列公式：

$$\text{vruntime} = \text{实际运行时间} * 1024 / \text{进程权重}。$$

实际上vruntime就是根据权重将实际运行时间标准化，标准化之后，各个进程对资源的消耗情况就可以直接通过比较vruntime来知道，比如某个进程的vruntime比较小，我们就可以知道这个进程消耗CPU资源比较少，反之消耗CPU资源就比较多。然后就调度vruntime较小的进程。

## 死锁，同步，互斥

- 死锁是指两个或两个以上的进程在执行过程中，由于竞争资源或者由于彼此通信而造成的一种阻塞的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。
- 同步就是两个及两个以上的操作就有严格是先后顺序，其中任何一个操作依赖于另外的一个或几个操作，例如课上的例子乘务员和司机，司机是否开车依赖于乘务员是否关车门。而异步是指，几个不同的操作之间没有绝对的依赖关系。
- 临界区：把不允许多个并发进程交叉执行的一段程序称作临界区。互斥：一组并发进程中的一个或多个程序段，因共享某一公有资源而导致它们必须以一个不允许交叉执行的单位执行不允许两个以上的共享该资源的进程同时进入临界区

## 互斥锁

- 在编程中，引入了对象互斥锁的概念，来保证共享数据操作的完整性。每个对象都对应于一个可称为"互斥锁"的标记，这个标记用来保证在任一时刻，只能有一个线程访问该对象。

## 乐观锁与悲观锁

- 乐观锁：就是很乐观，每次去拿数据的时候都认为别人不会修改，所以不会上锁，但是在更新的时候会判断一下在此期间别人有没有去更新这个数据。
- 悲观锁：顾名思义，就是很悲观，每次去拿数据的时候都认为别人会修改，所以每次在拿数据的时候都会上锁，这样别人想拿这个数据就会block直到它拿到锁。传统的关系型数据库里边就用到了很多这种锁机制，比如行锁，表锁等，读锁，写锁等，都是在做操作之前先上锁。
- CAS：CAS便是乐观锁技术，当多个线程尝试使用CAS同时更新同一个变量时，只有其中一个线程能更新变量的值，而其它线程都失败，失败的线程并不会被挂起，而是被告知这次竞争中失败，并可以再次尝试。  
CAS有3个操作数，内存值V，旧的预期值A，要修改的新值B。当且仅当预期值A和内存值V相同时，将内存值V修改为B，否则什么都不做。

## 虚拟内存与物理内存

- 物理内存，就是运行内存。就是指计算机的安装内存“通俗的讲就是内存条的大小”
- 虚拟内存，一个进程运行时都会得到一定的虚拟内存，32位系统下是4G，进程得到的这4G虚拟内存是一个连续的地址空间（这也只是进程认为），而实际上，它通常是被分隔成多个物理内存碎片，还有一部分存储在外部的磁盘存储器上，在需要进行数据交换。
  - 每次访问一个地址，都需要把虚拟内存转换为物理内存
  - 所有进程共享这整一块物理内存，每个进程只把自己目前需要的虚拟地址空间映射到物理内存上
  - 虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。

## 并发和并行

- 并发是多个任务交替使用CPU，同一时刻还是只有一个任务在跑，并行是多个任务同时跑

## 死锁

- 死锁产生的条件：
  - （一）互斥条件：一个资源一次只能被一个进程访问。即某个资源在一段时间内只能由一个进程占有，不能同时被两个或两个以上的进程占有。这种独占资源如CD-ROM驱动器，打印机等等，必须在占有该资源的进程主动释放它之后，其它进程才能占有该资源。这是由资源本身的属性所决定的。
  - （二）请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放。进程至少已经占有一个资源，但又申请新的资源；由于该资源已被另外进程占有，此时该进程阻塞；但是，它在等待新资源之时，仍继续占用已占有的资源。
  - （三）不剥夺条件：进程已经获得的资源，在未使用完之前不能强行剥夺，而只能由该资源的占有者进程自行释放。
  - （四）循环等待条件：若干资源形成一种头尾相接的循环等待资源关系。
- 解决方法
  - 银行家算法

## 分页

- 分页（英语：Paging），是一种操作系统里存储器管理的一种技术，可以使电脑的主存可以使用存储在辅助存储器中的数据。操作系统会将辅助存储器（通常是磁盘）中的数据分区成固定大小的区块，称为“页”（pages）。当不需要时，将分页由主存（通常是内存）移到辅助存储器；当需要

时，再将数据取回，加载主存中。相对于分段，分页允许存储器存储于不连续的区块以维持文件系统的整齐。分页是磁盘和内存间传输数据块的最小单位。