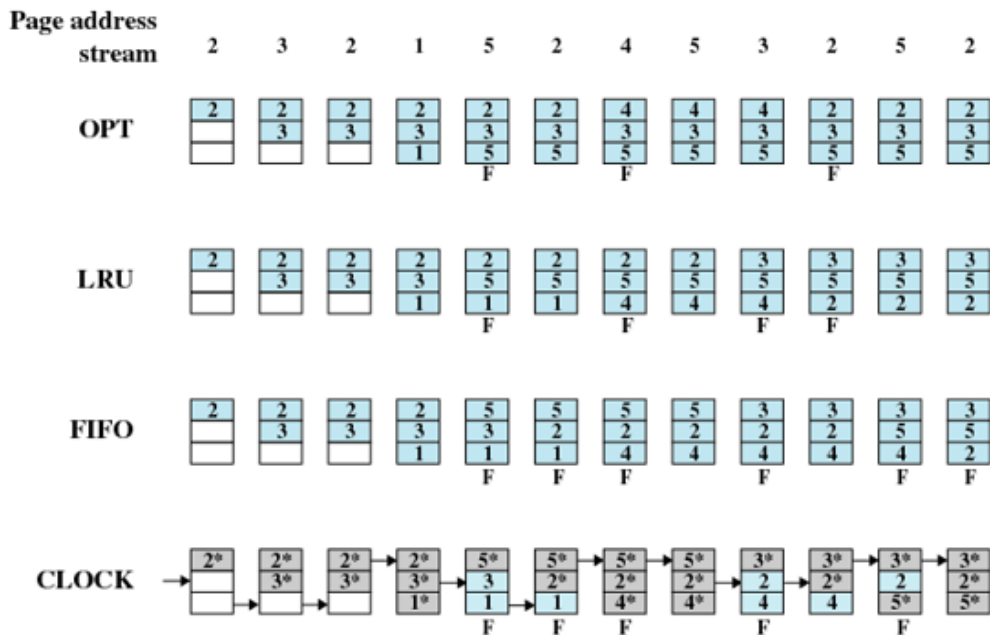


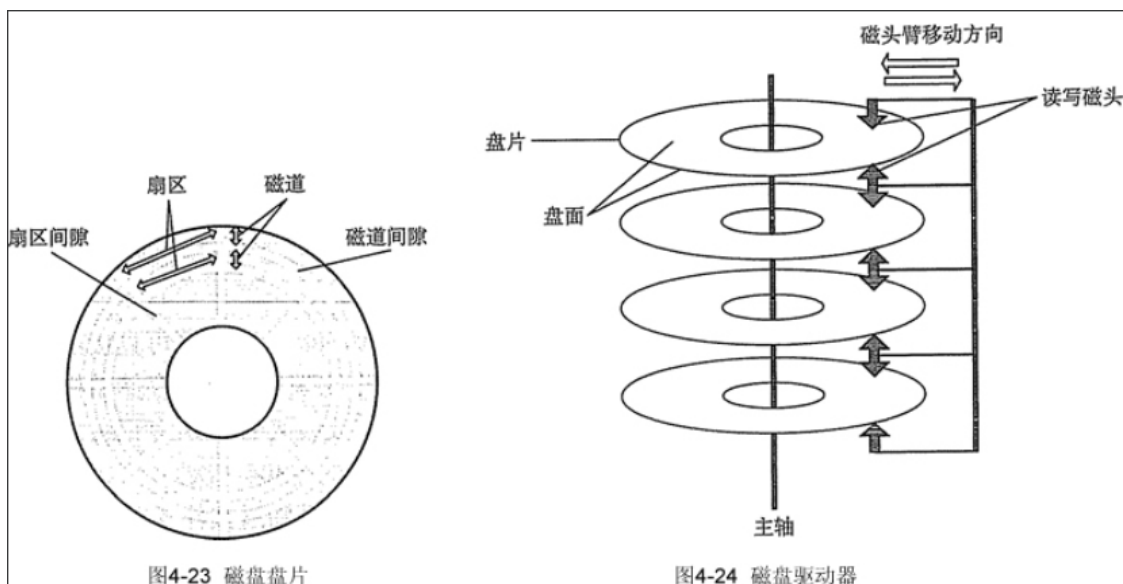
URL: https://blog.csdn.net/csp_6666/article/details/114606913

- 识别问题：内核态可以识别同一个进程里的其他线程，但是用户态不可以识别。若内核态中的进程A中的一个线程阻塞了，那么它可以从A的其他线程，以及其他进程的线程中任意选择；但是用户态就只能从其他进程的线程中选择。

- 特权级别：特权环：R0、R1、R2和R3，R0相当于内核态，R3相当于用户态；不同级别能够运行不同的指令集合；
 - CPU状态间的转换：用户态—>内核态：唯一途径是通过中断、异常、陷入机制（访管指令），内核态—>用户态：设置程序状态字PSW；
 - 功能：内核态负责运行操作系统程序，操作硬件；用户态负责运行用户程序。
- 竞争条件是什么？
 - 当多个进程或线程访问共享资源时，最终结果取决于多个进程中指令的执行顺序。
- 什么是死锁？（deadlock）
 - 一组竞争系统资源或互相通信的进程间相互的“永久”阻塞，没有有效的通用解决方案。
 - 死锁的4种解决方案：忽略、预防、避免、检测与破坏
 - 死锁的4个条件：互斥、占有且等待、非抢占、循环等待
- 什么是银行家算法？（Banker algorithms）
 - 银行家策略即资源分配拒绝策略。
 - 一个系统有固定数目的进程和资源，任何时候一个进程可能分配到0个或多个资源，该策略确保系统中进程和资源总是处于安全状态。当进程请求一组资源时，假设同意该请求，则改变了系统的状态，然后确定其结果是否还处于安全状态。如果是，同意这个请求；如果不是，阻塞该进程直到同意该请求时系统仍是安全的。
 - 安全状态是指至少有一个进程执行序列不会导致死锁。
 - 不安全状态是指一个不安全的状态，但不一定导致死锁。
- 什么是转移后备缓冲器？（TLB）translation lookaside buffer）
 - 每个虚存可能引起两次物理内存的访问。一次取相应的页表项，一次取需要的数据，这会导致访问时间加倍。
 - 为克服此问题，一个特殊的高速缓存为页表项使用，被称为TLB。
- 什么是“系统抖动”现象？
 - 进程运行过程中频繁地产生缺页中断，这种频率非常高的页面置换现象称为抖动。
 - 因此我们需要一个优秀的置换算法，降低缺页中断的频率。
- 4中替换算法分别是什么？（Page replacement algorithms，页面替换策略）
 - LRU，最近最久未使用：替换主存中上次使用距当前最远的页。
 - 根据局部性原理，这也是最近最不可能访问到的页
 - CLOCK，时钟置换算法：
 - 附加位称为使用位，初始时每个附加位都为 1
 - 当某一页首次装入主存时，该帧的使用位设置为 1，然后指针指向下一个使用位为 0
 - 若某一页已经在主存中，那么直接将对应的使用位设置为 1，然后指针指向下一个使用位为 0的地方
 - 当所有的使用位都为 1 时，我们将所有使用位都设置为 0，然后重新寻找下一个使用位为0 的地方
 - FIFO,优先队列算法：先进先出
 - OPT，最佳算法：接下来最晚被用到、或者不被用到的被替换

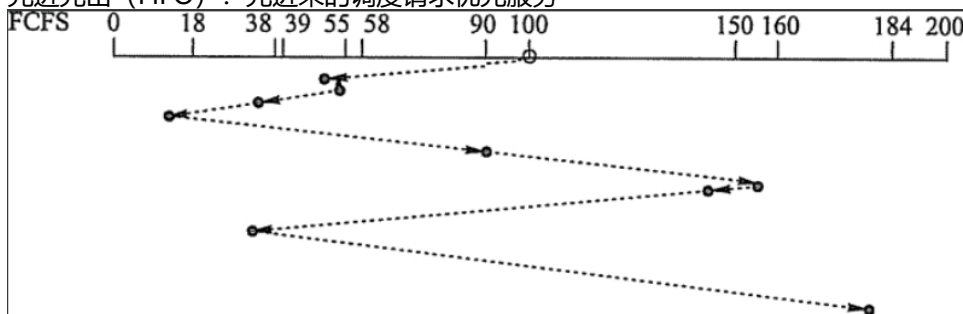


- 什么是抢占或者非抢占？ (preemptive or nonpreemptive)
 - 首先我们需要直到CPU调度决策可以在如下四种环境下发生：
 - 当一个进程从运行状态切换到等待状态（例如，I/O请求，或者调用wait等待一个子进程的中止）
 - 当一个进程从运行状态切换到就绪状态（例如当出现中断时）
 - 当一个进程从等待状态切换到就绪状态（例如I/O完成）
 - 当一个进程中止时
 - 抢占：当全部满足时，为抢占。在进程执行的过程中，可以被中断，然后调用其他程序。
 - 非抢占：当只有方案1, 4满足时，为非抢占。一旦进程处于运行状态，它就不断执行直到中止，或者因为等待I/O，或者因为请求某些操作系统服务而阻塞自己。
- 调度有哪些？ (schedule algorithms)
 - 远程调度：将进程加入进程池中
 - 中程调度 (mid-term)：给对应进程分配内存资源
 - 短程调度 (short-term)：将进程从就绪队列转换为运行队列
- 在短程调度中，什么是周期时间呢？ (turnaround time)
 - 周转时间 = 等待时间 + 服务时间
- 为什么磁盘缓存在操作系统中经常被用到呢？
 - 因为从磁盘中读取数据的速度远低于CPU的运行速度，一些写出的数据也许下次会被访问到，因此需要使这些数据能迅速地由软件设置的磁盘高速缓存中取出，而不是缓慢地从磁盘中取出。

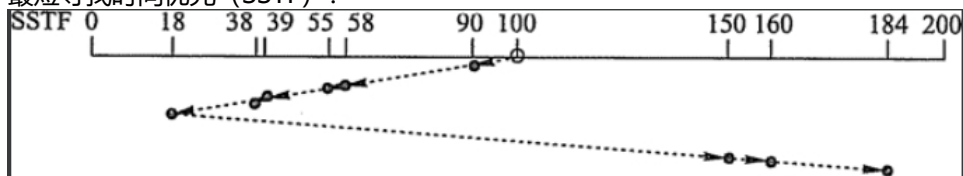


- 磁盘调度算法有哪些？

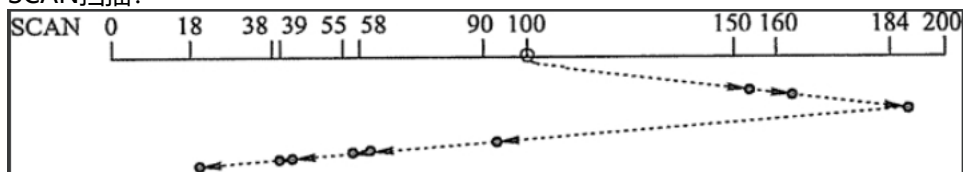
1. 先进先出 (FIFO)：先进来的调度请求优先服务



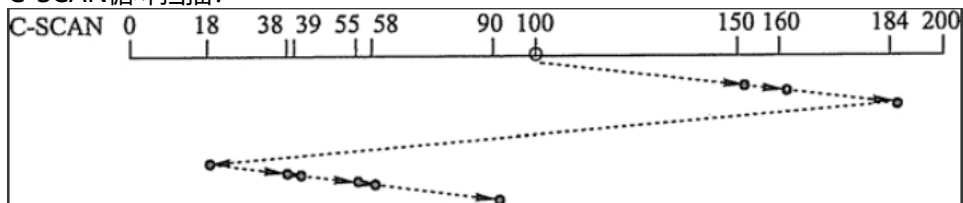
2. 最短寻找时间优先 (SSTF)：



3. SCAN扫描：



4. C-SCAN循环扫描：



- 什么是位视图法？

- 作用：在给文件分配空间时，是以磁盘的盘块为基本单位分配的，必须记录磁盘可用于分配的盘块(即空闲盘块)，以及提供磁盘分配和回收的手段。
- 利用二进制的一位来表示磁盘中的一个盘块的使用情况。当其值为“0”时，表示对应的盘块空闲；为“1”时，表示已经分配(或者把“0”作为盘块已分配的标记，把“1”作为空闲标志)。磁盘上的所有盘块都有一个二进制位与之对应，这样，由所有盘块所对应的位构成一个集合，称为位示图。通常可用 $m \times n$ 个数来构成 m 行 n 列的位示图，并使 $m \times n$ 等于磁盘的总块数。

- 什么是绝对路径和相对路径？(absolute and relative)

- 绝对路径是：以根目录为起点的路径
- 相对路径是：从当前目录为起点出发的路径
- 解释一下 /usr/ast/mbox 在Unix SVR4中是如何被解析的呢？

根目录	i-节点6	块132	i-节点26	块406
1 .	模式	6 .	模式	26 .
1 ..	大小	1 ..	大小	6 ..
4 bin	时间	19 dick	时间	64 grants
7 dev	132	30 erik	406	92 books
14 lib		51 jim		60 mbox
9 etc		26 ast		81 minix
6 usr		45 bal		17 src
8 tmp				

- 在根目录下找到节点6，读取其内容，发现usr是一个目录项；
- 从节点6的内容中读取到第一个块地址132
- 从块地址132中读取到节点26，发现ast是一个目录项
- 从节点26中读取到第一个块地址406
- 从块地址406中读取到节点60，发现它是一个文件，结束搜索
- 请问该文件大小是多少呢？

disk block size is 1 Kbytes:一块 1KB

4 bytes=4*8=32 位，所以一级间接是 2^{32} 块，二级则平方一次即可。

文件的容量如下表：

级	块数	字节数
直接	10	10KB
一级间接	$2^{32}=256$	256KB
二级间接	$256*256=64k$	64MB

