## 零散知识：

1. 进程运行控制通过使用 （1）进程创建 （2）进程撤销 （3）进程阻塞 （4）进程唤醒 等进程控制原语实现。
2. 程序运行过程中，只能进行wait操作、singnal操作的变量是：信号灯
3. 银行家算法是一种死锁避免。
4. 静态分配资源算法是一种死锁预防。
5. 较好解决内存碎片的内存管理方法：分页管理
6. 分页是为了解决非连续非配分配所产生的内存碎片问题，页是信息的物理单位。段是信息的逻辑单位，它含有一组意义相对完整的信息，分段的目的为了更好的实现共享，满足用户需求。
7. 一个页的大小是固定的，一个段需要多少整数倍的页，就请求多少个页，用完以后按照伙伴系统算法，返回到内存中，减少内存碎片。
8. 系统产生抖动的原因：淘汰算法选择不当。

在请求分页存储管理中，从主存（DRAM）中刚刚换出（Swap Out）某一页面后（换出到Disk），根据请求马上又换入（Swap In）该页，这种反复换出换入的现象，称为系统颠簸，也叫系统抖动。产生该现象的主要原因是置换算法选择不当。

危害：系统时间消耗在低速的*I/O上，*大大降低系统效率。进程对当前换出页的每一次访问，与对RAM中页的访问相比，要慢几个数量级。

1. 文件系统使用文件控制块管理软件

文件控制块：用于描述和控制文件的数据结构

1. 并发和共享是操作系统的两个最基本特征，两者互为存在条件。
2. 系统安全状态是找到一个进程结束的序列。
3. 分段管理中，逻辑地址是二维的，分别是段号和段内地址
4. 引入通道和中断技术的目的，是控制设备完成内存和外设之间的信息传送，提高I/O设备与CPU并行能力。
5. 计算机的管态，又称为系统态或核心态。
6. 分时系统是一个多用户交互系统，因此一定具有用户交互功能。
7. 在机器数中，补码和移码的值0的表现形式是唯一的。
8. 通过文件名存取文件时，文件系统内部的操作是通过：通过文件名找到对应的节点，通过节点存取文件数据。
9. 在Linux系统中，**线程**是处理器调度的基本单位，进程是拥有资源的基本单位。
10. 指令周期指
11. 是取指、分析指令、执行指令所需要的全部时间。
12. 指令周期大于机器周期
13. 虚拟内存唯一的决定因素是 计算机的地址位数
14. 计算机在多用户分时系统、多道批处理系统上，同一个程序多次运行，所花费的时间可能不一样。
15. 分配延迟：分配器停止一个进程到开启另一个进程的时间。
16. 通常与数据总线位数对应相同的部件是CPU
17. 分布式系统和网络系统的主要区别：透明性
18. 编码方案：
19. NRZ：高电平表示1，低电平表示0
20. NRZI：电平不变表示1，电平改变表示0
21. 请求分页系统中，页表的改变位用于记录该页表有没有被修改过，在该页被被换出的时候，如果该页表有被修改，就要把该页表写到硬盘上，如果没有，就把该页表丢弃
22. 同一进程中，线程共享进程的地址空间
23. 为了实现主机和外设之间信息传送，可以使用传送指令、输入/输出指令
24. 进程由程序、相关数据段、PCB组成
25. 常用的传输介质有3中：同轴电缆、双绞线、光纤
26. 内核线程：
27. 在内核支持下运行的
28. 用户线程：
29. 仅存在用户空间
30. 无需内核支持，内核完全不知道用户线程的存在。
31. renice指令可以改变进程的优先级
32. 文件系统在内存中维护唯一的一张分区表，其中保存了系统所有已经打开的文件的FCB。
33. 形成逻辑地址的阶段：链接阶段

## 多线程和多进程的区别：

1. 数据共享：多进程共享数据复杂，需要IPC（进程间通讯）；多线程共享进程的数据，数据共享简单。
2. 同步：多进程数据分开的（不同的进程数据在不同的用户空间），同步简单；多线程共享进程的数据，为了保证数据的安全性，同步比较复杂
3. 内存、CPU：多进程占用内存多，切换复杂，CPU利用率低；多线程占用内存少，切换简单，CPU利用率高
4. 创建、销毁：多进程创建、销毁速度慢；多线程创建、销毁速度快。（生成一个子进程，是父进程的一个复制品，生成一个线程相当于增加一个进程的执行流，公用进程的地址空间）
5. 分布式：多进程适用于多核、多机分布式；多线程适用于多核分布式

### 进程和线程的区别：

1. 进程是程序运行的一个实例，线程是一个控制执行流
2. 进程是资源分配的基本单位，线程是系统调度的基本单位
3. 进程拥有独立的地址空间，共享复制，需要使用进程间通讯（IPC），同步简单；线程共享所属进程的资源，共享简单，同步复杂，需要通过加锁等措施。
4. 进程内存不多，切换复杂，CPU利用率比较低；线程占用内存少，切换简单，CPU利用率高。
5. 进程间不会相互影响；一个线程挂掉可能整个进程挂掉。

## 虚拟页式存储置换算法及计算：

### LRU（最近最少使用算法）：

1. 如果未命中，将在缓存中，并且离（最后一个页号）最远的页置换出来。
2. 例题：某缓存系统采用LRU淘汰算法，假定缓存容量为4，并且初始为空，那么在顺序访问以下数据项的时候 1,5,1,3,5,2,4,1,2 出现缓存直接命中的次数是？，最后缓存中即将准备淘汰的数据项是？

LRU是最近最少使用算法，本题缓存中的页面替换过程如下：

1）1 进入缓存，缓存剩余容量为 3

2）5 进入缓存，缓存剩余容量为 2，最近最少使用为 1

3）从缓存中访问 1 ，出现 1 次直接命中，缓存剩余容量为 2 ，最近最少使用为 5

4）3 进入缓存，缓存剩余容量为 1， 最近最少使用为 5

5）从缓存中访问 5， 出现 2 次直接命中，缓存剩余容量为 1， 最近最少使用为 1

6）2 进入缓存，缓存剩余容量为 0，最近最少使用为 1

7）4 进入缓存，1 被淘汰，最近最少使用为 3

8）1 进入缓存，3 被淘汰，最近最少使用为 5

9）从缓存中访问 2， 出现 3 次直接命中，最近最少使用为 5

所以：

1）直接命中次数为 3；

2）即将被淘汰的是 5

# 进程调度算法及计算：

## 最高响应比优先：

1、响应比=（作业运行时间 + 等待时间）/ 运行时间