

- 信号量：信号量是一种与临界区资源相联系的变量；
- 线程：线程是指进程的一个可调度实体，是 CPU 调度的基本单位；
- Address mapping：将逻辑地址转为物理地址；
- PCB：进程控制模块，为描述进程的运动变化过程而引入的一个与进程相联系的数据结构，用于存储系统管理进程所需的信息，描述进程的瞬间特征；
- File system：文件系统指的是系统中负责存取和管理文件信息的机构；
- 中断：当某个事件发生时，系统中止现行的程序，引出处理程序对该事件进行处理，处理完毕后返回断点继续执行的过程；
- 进程：一个具有独立功能的程序关于某个数据集合的一次可以并发执行的运行活动，是操作系统进行资源分配和调度的一个独立单位；
- 虚拟内存：虚拟内存建立在架构之上，通过 cache 与 cpu 打交道，通过磁盘与用户打交道，他给外界的现象就是一个速度很快、容量很大的内存，由于这个不是物理上客观存在的，故称为虚拟内存；
- MMU：内存管理单元，负责将虚拟地址转为物理地址的设备元件；
- Thrashing：颠簸，在页式存储管理系统中，如果某一个或某些页面不停地从内存调出，然后又调入内存，这种频繁的页调度称为颠簸；
- 虚拟机：虚拟机是指通过软件模拟的、具有完整硬件系统功能的、运行在一个完全隔离环境的完整计算机系统；
- 死锁：某个进程所申请的资源被其他等待进程占有，那么该等待进程有可能再也无法改变其状态，这种状况成为死锁；

- Internal Fragmentation : 内部存储碎片, 已经分配出去的、不能被利用的内存空间;
- External Fragmentation : 外部存储碎片, 没有被分配出去的、内存大小无法满足申请内存的进程的内存空间区域;
- FCB : 文件控制模块, 为便于对文件进行管理和控制, 在文件系统内部给每个文件设置的数据结构;
- Caching : 高速缓存, 协调硬件之间速度差异的一种内存缓冲区, 是指存取速度比一般随机存取记忆体 (RAM) 来得快的一种 RAM。
- Race condition : 竞争条件, 多个进程并发访问和操作同一数据, 且执行结果与访问的特定顺序有关, 称为竞争条件;
- 忙等待: 某个进程循环检查某个条件是否为真, 且这个进程不会进入等待状态, 且不主动放弃 CPU 资源, 这种现象称为忙等待;
- 工作集合 (working set): 最近 dert (工作集合窗口) 个引用的页集合成为工作集合;
- Inode : 索引节点, 用来存档案及目录的基本信息的结构体;
- Cache coherency : 缓存一致性, 保持高速缓存中的共享资源的数据一致性的机制;
- 短期调度: 从准备执行的进程中选择进程, 并为之分配 CPU;
- 长期调度: 从缓冲池中选择进程, 并装入内存以准备执行;
- 中期调度: 将进程从内存或 CPU 竞争中移出, 从而降低多道程序设计的程度, 之后进程能够重新调入内存, 并从中断处继续执行;
- VFS : 虚拟文件系统, 是物理文件系统与服务之间的接口层, 是一种用于网络环境的分布式文件系统, 是允许和操作系统使用不同的文件系统实现的接口;

- DMA : direct memory access , 直接存储器访问 , 是所有现代电脑的重要元件 , 它允许不同速度的硬件装置相互沟通 , 而不需要依赖于 CPU 的大量中断负载 ;
 - Critical : 临界区 , 指的是一个访问共用资源的程序片段 , 且这些共用资源又无法同时被多个线程访问 ;
 - 临界区问题解答的三个条件 : 互斥 , 前进 , 有限等待 ;
 - 操作系统 : 是作为计算机硬件和计算机用户之间的中介的程序 ;
 - API : Application Programming Interface , 应用程序编程接口 , 是一些预先定义的函数 , 目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力 , 而又无需访问源码 ;
 - 系统调用 : 为管理硬件资源和为应用程序开发人员提供良好的环境来使应用程序具有更好的兼容性 , 内核提供一系列具备预定功能的多内核函数 , 通过一组称为系统调用的接口呈现给用户。系统调用把应用程序的请求传给内核 , 调用相应的的内核函数完成所需的处理 , 将处理结果返回给应用程序 ;
 - 内核 : 物理硬件之上、系统调用接口之下所有部分称为内核 ;
 - 微内核的优点 : 代码量少、易开发扩展、bug 数量少、更可靠安全 ; 缺点 : 速度慢、要进行两次系统调用 ;
 - 通信的两种模式 : 共享内存 , 消息传递 ;
- 消息传递 :
- 优点 : 可以用作同步机制来处理进程间的通信 ; 交换的数据量少时 , 不必避免冲突 ;
- 比共享内存更容易实现 ;
- 缺点 : 消息传递通常包含系统调用。

共享内存：

优点：允许通信的最大速度和方便性；进程间通信不需要内核的协助；

缺点：在保护和同步方面仍存在一定问题；没有提供协调通信进程同步的进程。

- 进程不能从 waiting 到 running：可能会增加调度算法的冗余度，且此时 CPU 可能在执行其他进程，可能会产生意外错误；
- 进程不能从 ready 到 waiting：只有执行之后系统才会直到需不需要请求 I/O；
- 上下文切换：context switch，将 CPU 切换到另一个进程需要保存当前进程的状态并恢复另一个进程的状态；
- 用户线程与内核线程的区别：用户线程受内核支持而无须内核管理；而内核线程由操作系统直接支持和管理；用户线程只能通过内核线程请求 I/O；内核线程创建需要较多资源，用户线程方便创建；
- 多对一：效率高；并发性差，一个阻塞则整个进程阻塞；一对一：并发性好、不会阻塞；创建一个用户线程就要创建一个内核线程，开销大；
- 线程池优点：用现有的线程处理请求比创建新的线程块；线程池限制了任何时候可用线程的数量，对不能支持大量并发的系统非常重要；
- 死锁的四个条件：互斥、占有并等待、非抢占、循环等待；
- MMU 通过基地址（base）寄存器和界限地址（limit）寄存器将逻辑地址转为物理地址并保护其免受用户线程的修改：

转：基地址寄存器这里推广为重定位寄存器，逻辑地址进入重定位寄存器后加上重定位寄存器的值后转为物理地址；

保护：只有操作系统可以通过特殊的指令加载基地址寄存器和界限地址寄存器，由于特

权指令只能在内核模式下运行且只有操作系统可以再内核模式下运行，故只有操作系统可以加载基地址寄存器和界限地址寄存器，故用户线程无法修改他们；

- 动态存储分配：首次适配（第一个可以的）、最佳适应（可以的里面最小的）、最差适应（最大的）；
- 静态页面管理（paging）将一个作业的全部地址空间都放入内存中，请求式页面管理（demand paging）则按照需求将页面的对应地址空间放入内存中
- 发生页面错误：检查进程的内部页表确定该引用是否合法，合法但尚未调入页面则从内存中调入，找一个空闲帧（无空闲帧时按照一定的置换算法将某一页置换出来腾出空间），调度磁盘操作将所需要的页调入刚分配的帧，修改进程的内部表和页表，重新开始因陷阱而中断的指令。
- 写时复制：父进程地址空间的复制可能没有必要。允许父进程与子进程开始时共享一个页面，这些页面标记为写时复制页，如果任何一个进程需要对页进行操作，那么就创建一个共享页的副本；
- 颠簸的原因：如果分配给进程的存储块数量小于进程所需要的最小值，进程的运行将很频繁地产生缺页中断，这种频率非常高的缺页现象导致了颠簸；
- 文件分配方法：连续分配、链接分配、索引分配；
-