操作系统

- 操作系统的基本特征：**并发、资源共享**
  - 资源：硬件资源、信息资源
- **操作系统提供给用户的界面**
  - GUI, 图形用户界面
  - 批处理
  - 命令行
- 中断：外部中断、内部中断（陷入）、异常
- 操作系统双重模式：内核模式、用户模式
- 系统调用类型：
  - Process control 进程控制
  - File manipulation 文件管理
  - Device manipulation 设备管理
  - Information maintenance 信息维护
  - Communications 通信
- 操作系统分为：微内核、宏内核

临界资源

一次仅允许一个进程使用的资源称为临界资源。

- 物理设备，如打印机。
- 被若干进程共享的变量、数据

临界区

每个进程中访问临界资源的那段代码称为临界区。

- 解决临界区问题：**互斥、前进、有限等待**

访问临界资源的原则

- 如果有若干进程要求进入空闲的临界区，一次仅允许一个进程进入。
- 任何时候，处于临界区内的进程不可多于一个。
- 进入临界区的进程要在有限时间内退出，以便其它进程能及时进入自己的临界区。
- 如果进程不能进入自己的临界区，则应让出CPU，避免进程出现“忙等”现象。

PCB是进程存在的唯一标志

PCB里包含

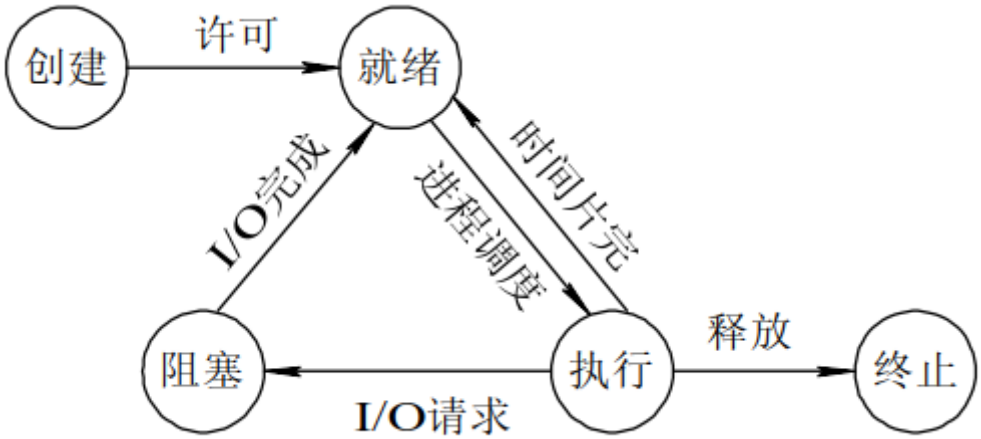
- 进程状态
- 程序计数器
- CPU寄存器
- CPU调度信息
- 内存管理信息

- 记账信息
- I/O状态信息

进程间通信方式

- **共享内存：** 仅在建立共享内存区域时需要系统调用，一旦建立了共享内存，所有的访问都被处理为常规的内存访问，不需要来自内核的帮助
- **消息传递：** 通常用 ***系统调用*** 来实现，因此需要更多的内核介入的时间消耗
- **管道通信：** 共享内存的优化和发展，讲存储空间进化成了缓冲区。

进程状态的状态和转换



进程和线程的联系

- 一个线程只能属于一个进程，而一个进程可以有多个线程，但至少有一个线程。
- 资源分配给进程，同一进程的所有线程共享该进程的所有资源。
- 处理机分给线程，即真正在处理机上运行的是线程。
- 线程在执行过程中，需要协作同步。不同进程的线程间要利用消息通信的办法实现同步。线程是指进程内的一个执行单元,也是进程内的可调度实体.

进程和线程的区别

- **调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位**
- 并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可并发执行
- 拥有资源：进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源.
- 系统开销：在创建或撤消进程时，由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销。

线程

- 线程与属于同一进程的其他线程共享：代码段、数据、文件
- 线程模型：多对一、一对一、多对多

多线程的优势

- 高响应度：如果对一个交互程序采用多线程，那么即使其部分阻塞或执行较冗长的操作，该程序仍能继续执行，从而增加了对用户的响应程度
- 资源共享：线程默认共享它们所属进程的内存和资源。代码和数据共享的优点是它能允许一个应用程序在同一地址空间有多个不同的活动线程
- 经济：进程创建所需要的内存和资源的分配比较昂贵。由于线程能共享它们所属进程的资源，所以创建和切换线程会更为经济
- 多处理器体系结构的利用：多线程的优点之一是能充分使用多处理器体系结构，以便每个进程能并行运行在不同的处理器上。不管有多少CPU，单线程进程只能运行在一个CPU上。在多CPU上使用多线程加强了并发功能

死锁：多个进程因竞争资源而造成的互相等待，若无外力作用，它们都将无法推进下去。

死锁产生原因

- 系统资源的竞争。对不可剥夺资源的竞争。
- 进程推进顺序非法。
    - 请求和释放资源的顺序不当
    - 信号量使用不当
- 死锁产生的必要条件
    - **请求并保持**：一个进程必须占有至少一个资源，并等待另一个资源，而该资源为其他程序占有
    - **互斥**：至少有一个资源必须处于非共享模式，即一次只有一个进程使用。如果另一个进程申请该资源，那么申请进程必须等到该资源被释放为止。
    - **不剥夺**：资源不能被抢占，即资源只能在进程完成任务后自动释放
    - **循环等待**：有一组等待进程｛P0，P1，……，Pn｝，P0等待的资源为P1所占有，P1等待的资源为P2所占有，……Pn - 1等待的资源为Pn所占有，Pn等待的资源为P0所占有。

解决死锁

- **死锁预防**：破坏上文死锁的四个特征条件之一即可，但这种破坏并不合理
- 死锁避免：寻找安全序列——银行家算法
- 死锁检测+死锁恢复：死锁检测采用银行家算法

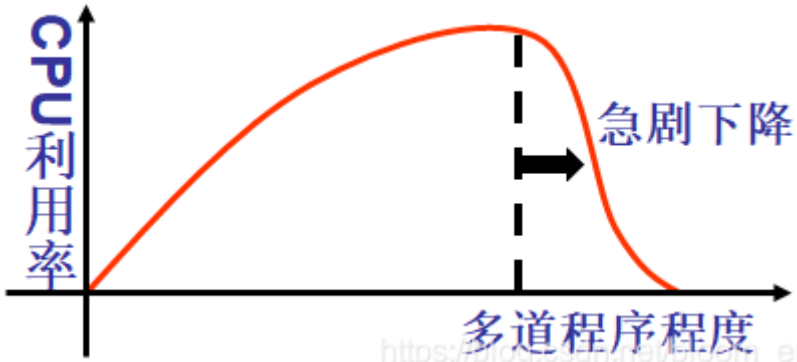
其他

- 上下文切换
    - 内核会将旧进程的状态保存在其PCB中
    - 而后内核后装入经调度要执行的并已保存的新进 程的上下文
- 重定位
    - 定义：重定位就是把程序的逻辑地址空间变换成内存中的实际物理地址空间的过程。
    - 编译时：如果编译时就知道进程在内存中的驻留地址，那么就可以生成绝对代码——即生成绝对地址，不可更改
    - 加载时：如果编译时不知道进程将驻留在内存的什么地方，那么编译器就必须生成可重定位代码。
    - 运行时：如果进程执行时可以从一个内存段移动到另一个内存段，那么绑定必须延迟到执行时才进行。
    - **重定位最合适的时机**：运行时重定位（基地址加偏移量）

- MMU（内存管理单元）
  - 作用：完成从虚拟地址到物理地址的映射，将虚拟地址送往总线之前转换成物理地址
  - 为什么使用：地址绑定在运行时重定位，每次都需要根据基地址和偏移地址计算物理地址，这一个过程采用硬件的方式速度更快。
- 碎片
  - 外部碎片：内存块加起来能满足一个请求，但是由于不连续（中间有断层），不能用来连续分配
    - 解决方案：紧缩、分页、分段
  - 内部碎片：分出去的分区略大于请求的内存长度。这个余下的小内存块属于该分区，但是无法利用
- **页表很大的解决**
  - 层次页表/多级页表
    - 实现思路
      - 将页表再分页，将页表的逻辑地址拆分成多张页表
      - 顶层页表常驻内存，不需要映射的逻辑地址不需要建立页表项
    - 特点：提高空间利用率
  - 哈希页表
    - 实现思路
      - 虚拟地址（逻辑地址）的页号经过哈希函数转换后，指向页表中某个页表项
      - 哈希函数值相同的虚拟页号，指向同一个页表项，它们在那个页表项下组成一个链表
      - 地址翻译时，由虚拟页号哈希后锁定对应链表，搜索链表得到虚拟页号的匹配项
      - 如果找到匹配项，则找到了虚拟页号对应的物理页帧
    - 特点
      - 当一个程序占用大的虚拟地址空间的一小部分时，哈希页表非常合适。
      - 但页面较多会引起冲突，导致其相应的哈希入口负担过重，从而在链表上搜索的代价非常高。
  - 反向页表
    - 实现思路
      - 为每一个物理块设置一个也表项并按物理块号排序，其中的内容则是页号及属于的进程的标志符。
      - 整个系统只有一个页表，对每个物理内存的页只有一条相应的条目。
    - 特点：
      - 减少页表占用的内存空间
      - 有可能需要查找整个表来寻求匹配，增加了查找页表所需要的时间

CPU抖动

- 现象：页面刚换入然后又立即被换出
- CPU利用率急剧下降的原因
  - 系统内进程增多，导致每个进程的缺页率增大
  - 缺页率增大到一定程度，进程总等待调页完成，而调页需要用到磁盘
  - 导致一直都在等待，CPU利用率降低
- 防止手段：给进程分配足够多的帧



一些知识点

1. 操作系统是对计算机资源进行管理的软件
2. 从用户的观点看，OS是用户与计算机之间的接口
3. OS提供给程序员的接口是系统调用（库函数是高级汇编语言提供的，不是OS提供的）
4. 批处理系统的主要缺点是缺少交互性（交互性是分时系统引入的目的，正是因为批处理系统缺少交互性）
5. 虚拟机（Virtual Machine）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中的完整计算机系统。虚拟机的一个本质特点是运行在虚拟机上的软件被局限在虚拟机提供的资源里——它不能超出虚拟世界。对于用户而言，它只是运行在物理计算机上的一个应用程序，但是对于在虚拟机中运行的应用程序而言，它就是一台真正计算机。
6. 进程是程序的一次执行过程，是动态的．它有生命周期。这些都是进程的基本特性，进程是由程序段，数据段和PCB三部分组成。程序才是指令的集合，而进程不是。
7. 并发进程是指宏观上可同时执行的进程
8. 保存在PCB中的有进程标识符信息，处理机状态信息（寄存器），进程调度信息（进程标识符、进程当前状态、代码段指针、通用寄存器值）
9. 一个进程P被唤醒后，P的状态变成就绪
10. 若把操作系统看作资源管理者，中断不属于操作系统所管理的资源。（中断不是资源）

局部性原理相关

最基本的两种分类

- 时间局部性原理：如果执行了程序中的某条指令，那么在不久以后这条指令很有可能会再被执行；如果访问了某个数据，那么在不久以后这个数据很可能再被访问
- 空间局部性原理：如果访问了某个存储单元，在不久以后，其附近的存储单元也有可能被访问

高速缓冲技术的思想：将近期会频繁访问到的数据放进更高速的存储器中，暂时用不到的数据放到更低速的存储器中

局部性原理的应用：

- cache
- TLB
- 现实生活中书包和书架