操作系统

1 操作系统综述

1. 操作系统 控制计算机系统软硬件集合的系统，是计算机中最基本的系统软件
2. 基本特征 并发（同一时间间隔） 共享 虚拟（时分复用（处理器分时共享），空分复用（虚拟存储器）） 异步（不可预知的方式向前推进）
3. 功能 1.系统资源的管理者 2.用户和计算机硬件之间的接口（命令接口（组织和控制作业的执行（无需用户操心），分为联机和脱机两种）和程序接口（系统调用）） 3.作为扩充机器，控制裸机
4. 库函数是语言或者程序的一部分，运行在用户空间中，而系统调用是操作系统的一部分，运行在内核空间中，许多库函数都会调用系统调用函数来实现；使用了系统调用的库函数执行效率要低于不使用系统调用的，因为在使用系统调用时需要注意上下文的切换
5. 操作系统的发展 手工操作->单道批处理->多道批处理（并行）->分时（人机交互）->实时操作系统->分布式操作系统（协同完成同一个任务）
6. 分时系统最重要的是响应时间，采用 **优先级+非抢占式调度算法** 可以让重要的进程优先获得系统响应，又可以让其他进程不会得不到系统响应
7. 甘特图 横轴时间，竖轴进程
8. 操作系统的运行机制 1.时钟管理 2.中断机制 3.原语（底层，原子性，运行时间短）
9. 当中断或者异常发生时，运行在用户态的CPU会立刻进入核心态，这是通过 **硬件** 实现的
10. 中断分为外中断和内中断（异常）；外中断指来自于CPU执行指令之外的事件的发生；异常指CPU执行指令发生的中断，比如程序的非法操作码，地址越界，算术溢出，缺页，陷入等
11. 从用户态切换到核心态，不仅仅是状态的切换，使用的堆栈也要切换为系统堆栈，这个系统堆栈也是进程的，由硬件完成转换
12. 从用户态切换到核心态，会用到访管指令，访管指令是在用户态下使用的，所以访管指令不是特权指令；中断处理的程序工作在内核态，属于操作系统程序
13. 从用户态切换到核心态，的中断是访管中断
14. 编译器属于软件，不是操作系统必须要提供的
15. 通道技术是一种硬件技术
16. 中断是操作系统必须提供的技术
17. 命令解释程序属于命令接口
18. 系统调用指令（广义指令）工作在核心态，“调用”可以从用户态和核心态；但执行一定在“核心态”
19. 微内核 精简内核，降低了内核设计的复杂性，但性能不好（需要不停的切换用户态和核心态），稳定

2 进程管理

1. 进程 描述和控制进程的并发执行；进程控制块（PCB） 配置一个进程的专门的数据结构；进程映像 由PCB，程序段，相关数据段构成的
2. PCB是进程存在的唯一标识
3. 进程是动态的，进程映像是静态的
4. 进程的特征 动态性（根本特征） 并发性 独立性（一个进程实体是一个独立获得资源的基本单位） 异步 结构性
5. 进程的状态转换
    - 就绪状态 只缺少处理机，资源已经满足
    - 需要注意：进程创建不成功，是因为资源不足，此时进入阻塞状态
    - 阻塞是被阻塞进程自我调用实现的（自己资源不够自己知道）；唤醒是其他进程使用完资源将它唤醒的
6. 进程间的通信 共享存储（共享空间） 消息传递 管道通信（一次性，半双工，不受磁盘大小的限制，管道满时写阻塞，管道空是读阻塞）
7. 线程是CPU分配的基本单元，线程不拥有系统资源，使用进程的；线程没有自己的地址空间
8. 多线程模型 多对一（易阻塞） 一对一（开销大） 多对多
9. 易错点 单处理器中，任何时刻都只有一个进程处于运行状态是错的；有可能都处于阻塞状态
10. 进程在处理器中执行时，有时候是无关的，有时候是有联系的
11. 在多对一模型中，线程的切换不需要引起内核线程的切换，所以不需要内核的支持
12. 二进制代码和常量放在正文段，动态分配的存储区放在堆，局部变量放在栈
13. 易错点 设备分配是在系统中设置相应的数据结构而实现的，不需要创建进程
14. 处理机调度 从就绪队列中按照一定的算法分配处理机
    - 作业调度（创建->就绪，获得竞争处理机的权利，多道批处理中有，其他系统大多不需要有）；中级调度（阻塞（挂起）->就绪，提高系统吞吐量）；进程调度（低级调度，就绪->运行，最基本）
    - 不进行进程调度的情况 中断，临界区（不破坏临界区的规则可以调度），原子操作
    - 调度的基本准则
        - 系统吞吐率 单位时间内CPU完成作业的数量
        - 周转时间 作业提交到作业完成的时间
        - 带权周转时间 周转时间/作业实际运行时间
        - 响应时间 第一次响应的时间
    - 调度算法 FCFS（有利于CPU繁忙型，不利于I/O繁忙型） SJF（短作业优先，会导致饥饿） 优先级调度（还可用于作业调度） 高响应比优先（响应比=等待时间/要求服务时间） 时间片轮转法 多级反馈队列调度法
15. 分时系统、时间片轮转法相关的 都是为了多用户能够及时响应
16. 设置进程优先级时，I/O型作业的优先权高于CPU型，即IO时间长的作业应该先被执行
17. 进程同步
    - 临界资源 一次只允许一个进程访问的资源
    - 临界区 访问临界资源的代码

- 互斥进程的同步机制 空闲让进，忙则等待，有限等待，让权等待（和优先级无关，当进程不能进入临界区时，立即释放处理器，防止进程忙等待）
- 软件实现临界区互斥
  - 单标志法 turn=0 0表示临界区的进程编号
  - 双标志法先检查 while(flag[j]); flag[i]=true；i表示临界区进程编号，可能导致都进入临界区
  - 双标志法后检查 flag[i]=true; while(flag[j]); 可能导致饥饿，两个进程都进不去

  +peterson's 算法 flag[i]=true; turn=j; while(flag[j] && turn==j)
- 硬件实现临界区互斥
  - 中断屏蔽（关中断），是进程不可以被打断
  - Test and Set 原子操作，使用这个操作不断检查临界资源的状态，破坏了让权等待原则
- 信号量机制
  - 整型信号量 wait(S),P signal(S),V 低级操作原语，不是系统调用 阻塞型，未遵循让权等待
  - 记录型信号量 value=1，有资源可用；value=0，无阻塞，所有资源被使用；若+1之后value<=0，说明还有等待该资源的进程被阻塞，应该使用wakeup唤醒其他进程
  - 遵循让权等待机制
  - 同步操作，先V后P；互斥变量使用P V包起来
- 管程 局部于管程的数据只有被局部于管程内的过程所访问；每次仅允许一个进程在管程内执行某个内部过程；无需程序员关注；可以实现同步和互斥
- 公共队列属于临界资源

18. 死锁 多个进程竞争资源而造成的一种僵局（互相等待）
    - 死锁产生的原因 1.系统资源的竞争，不可剥夺资源 2.进程推进顺序非法
    - 死锁产生的必要条件 互斥 不可剥夺 请求与保持 循环等待
    - 死锁的预防 破坏死锁的必要条件 破坏请求与保持的方法：一次性分配
    - 死锁的避免 在资源动态分配中，防止系统进入不安全状态（银行家算法）
    - 死锁的检测 资源请求图（框：资源，圈：进程，框中的圈：资源数量，进程到资源：请求边，资源到进程：分配边）
    - 死锁的解除 1.资源剥夺法 2.撤消进程法 3.进程回退法
    - 并非所有的不安全状态都是死锁状态，但当系统进入不安全状态之后，可能进入死锁状态

3 内存管理

1. 创建进程需要将程序和数据装入内存，将源程序变成可在内存中执行的程序，需要通过编译、连接、装入（内存）

- 链接 静态链接（先链接再装入） 装入时动态链接 运行时动态链接
- 装入 绝对装入（绝对地址） 可重定位装入（静态重定位，实际上是在链接时生成逻辑地址，装入时加上一个偏移量，要有足够的内存空间） 动态运行时装入（动态重定位）（装入程序把模块装入内存时，不立即把装入模块的相对地址转换为绝对地址，而是在运行的时候转换）

2. 地址重定位 当装入程序将模块装入内存时，将相对地址转换为绝对地址
3. 内存保护 通过采用重定位寄存器（基址寄存器）和界地址寄存器（限长寄存器），界地址防止地址越界，重定位用来计算物理地址；实现现场保护需要保存这两个地址
4. 覆盖 同一进程有覆盖区；交换 不同进程交换
5. 连续分配管理方式 为程序分配一个连续的内存空间
    - 单一连续分配 分为系统区（低地址）和用户区，用户区永远只有一道程序；适合于单用户、单任务，有内部碎片
    - 固定分区分配 多道程序存储管理方式，有分区大小相等和不等两种分法，有内部碎片
    - 动态分区分配 外部碎片（通过紧凑解决） 首次适应（空闲分区按地址由小到大排序，最简单，最快，最好） 最佳适应（按容量大小排序） 最坏适应（按容量由大到小排序） 邻近适应（循环首次适应）
6. 非连续分配管理方式
    - 分页存储 分区大小固定，分为基本分页式（一次性装入内存）和请求分页式（可替换）
        - 每个进程平均产生半个块大小的内部碎片（页内碎片）
        - 进程中的块称为页，内存中的块称为页框（页帧），外存中的块就叫块
        - 分页存储管理的逻辑地址结构 页号+页内偏移量
        - 页表 页表项的组成：页号+块号
        - 页表项的块号和逻辑地址的页内偏移量组成物理地址
        - 页式管理中地址空间是一维的
        - 加入快表TLB（联想寄存器），基于局部性原理
        - 多级页表的顶级页表最多只能有1个页
    - 基本分段式
        - 段内要求连续，段间不连续
        - 地址结构 段号+段内偏移量
        - 段表 段号+段长+本段的起始地址
        - 分段方式的地址空间是二维的
    - 段页式 先分段再分页
        - 地址结构 段号+页号+页内偏移地址
        - 段表最多只能有1个
        - 先通过段号在段表中查找页表的起始地址，再通过页号在页表中查找物理块号，和偏移地址结合就是物理地址
7. 形成逻辑地址的阶段是链接；将逻辑地址转换为物理地址的过程是装入

8. 进程正在IO操作时不能会换出主存，否则它的数据会被其他进程清空
9. 页表的始地址存放在寄存器中
10. 程序的动态链接和逻辑结构有关，因此分段存储有利于程序的动态链接
11. 对主存的访问是以块为单位
12. 多进程系统中，所有进程的页表都存储在内存中，系统中有页表寄存器（PTR）,其中存放的是页表的起始地址和页表的长度
13. 虚拟内存管理 局部性原理
    - 请求分页式
      - 页面置换方式 最佳置换（OPT，以后最长时间不被使用的页面被淘汰） 先进先出（FIFO，会出现Belady异常，效率随着分配的物理块数量不减反增） 最近最久未使用（LRU，用以前估计未来） 时钟置换（CLOCK，最近未用，NRU）
    - 驻留集大小的分配策略 固定分配局部置换 可变分配全局置换（有一个救火的空闲物理块队列） 可变分配局部置换（系统可以减少给进程分配的物理块）
    - 从何处调入页面 拥有足够的对换区，全部从对换区调入；没有足够的对换区空间，反不会被修改的文件从文件区调入，会被修改的从对换区调入；UNIX中，未运行的页面从文件区调入，运行过的页面从对换区调入
14. 抖动（颠簸） 频繁的发生缺页中断
15. 工作集 在某段时间间隔内，进程要访问的页面集合
16. 虚存的大小要同时满足：1.小于内存和外存容量之和 2.小于计算机的地址位数能容纳的最大容量
17. 页表项中的合法位表示是否在本页是否在内存中
18. 快表用于存储计算机中交换最频繁的页面，用于地址交换
19. 工作集 最近k次内存访问所访问的页面的集合
20. 覆盖技术和虚拟存储技术的最本质的不同在于程序段的最大长度要受内存容量大小的限制，而虚拟存储器的大小征收计算机地址结构的限制

4 文件管理

1. 文件的基本操作 创建、查找、删除、写、读、截断（将文件长度直接设为0）
2. 文件的打开 首次使用文件的时候，调用系统调用open，从外存拷贝到内存打开文件，并将文件的信息保存到打开文件表（文件指针，文件打开计数，文件磁盘位置，访问权限等）；注意open并不是指向文件的指针，只是一个系统调用命令，指针存在打开文件表中
3. 文件的逻辑结构 从用户理解的角度出发
    - 无结构文件（流式文件） 以字节为单位，比如源代码文件，目标代码文件等
    - 有结构文件
      - 顺序文件 串结构（无序） 顺序结构（有序） 批量操作，顺序文件的效率是最高的，但对于增删查改操作比较困难
      - 索引文件 索引表（本身是一个定长记录的顺序文件），索引文件每个记录的大小不是定长的

        - 索引顺序文件 为索引表中每组的第一个文件做索引记录 $\sqrt{N}$
        - 索引文件 将键值使用Hash函数转换成记录的物理地址

4. 文件的目录结构
    - 文件控制块 存放控制文件需要的各种信息的数据结构，一个FCB就是一个文件目录项；包含基本信息，存取控制信息，使用信息等；FCB必须连续存放
    - 索引结点 如果使用FCB直接放在目录项中查找文件，FCB中太多无关的信息使文件目录项太大，所以使用只包含FCB中的文件指针（指针指向索引结点，里面有文件的相关信息），节省空间
    - 单级目录结构 查找速度慢，文件不允许重名，不利于文件共享
    - 两级目录结构 按用户划分，解决不同用户可以有重名文件
    - 多级目录结构 绝对路径
5. 文件共享 硬链接查找速率大于软链接
    - 硬链接 基于索引结点的共享方式 创建的链接指向文件的索引结点；删除时候只减少计数器，并删除自己文件目录的索引结点指针
    - 软链接（符号链接） 创建的链接指向A索引结点指针 A可直接删除索引结点指针，此时B无法找到这个文件
6. 文件保护 口令保护、加密保护、访问控制保护（控制用户访问权限）
    - 访问控制列表（Access Control List，ACL），规定每个用户可以访问的类型
7. 目录文件存放的是此目录中所有子目录文件和数据文件的信息
8. 建立符号链接时，计数器直接复制；建立硬链接，计数器+1
9. 两个进程通过索引结点共享一个文件，但是文件描述符由各自的进程保存
10. 文件系统的物理结构（如何实现）
    - 用户调用接口（open，read等）->文件目录系统（查找文件目录项中的索引结点指针）->存取控制验证（有没有权限操作）->逻辑文件系统和文件换存区（找到逻辑记录的块号）->物理文件系统（将逻辑记录转化为物理记录）->分配模块（分配空闲空间和回收）->设备管理程序模块（分配设备，磁盘调度，启动设备，中断等）
    - 文件分配方式
        - 连续分配 分配连续空间，支持顺序访问和直接访问；实现简单，存取速度快，但文件长度不宜动态增加
        - 链接分配 隐式链接，目录系统中的表项包含文件第一块的指针和最后一块的指针，每个盘块最后都有下一个盘块的地址（顺序访问，不安全可能会断掉，占有格外空间）；显式指针，使用文件分配表将所有盘块号对应的指针标出，文件分配表存放在内存中
        - 索引分配 将所有块的链接放在一个盘块中，目录系统的表项保存这个盘块的地址
    - 外存空间管理 空闲表法 空闲链表法 位示图法 成组链接法
11. 磁盘 磁道 扇区（一个扇区称为一个盘块） 磁道间隙 扇区间隙 磁头（固定头：每个磁道固定一个磁头；活动头：磁头臂可以来回移动寻道）
12. 磁盘的存储能力受限于最内道的最大记录密度
13. 磁盘读写时间：

- 寻道时间 $T_s = mxn + s$
- 延迟时间（寻找扇区） $T_r = \frac{1}{2r}$ r：磁盘旋转速度
- 传输时间 $T_t = \frac{b}{rN}$ b：读写的字节数；r：转速；N：每个磁道容纳的字节数
- 上诉三部分时间只有传输时间不能被缩短

14. 磁盘调度算法 FCFS SSTF（最短寻找时间优先） 扫描算法（Scan，电梯算法） 循环扫描（C-Scan，返回时返回到最开始）
15. 磁盘初始化
    - 低级格式化 将磁性记录材料的空白盘分为扇区，头部和尾部包含特殊的控制信息
    - 逻辑x格式化 分区和创建文件系统（FAT等）
16. 引导块 自举程序，启动操作系统之前，初始化CPU、寄存器、设备控制器和内存等，找到操作系统内核装入内存；装有启动分区的磁盘称为启动磁盘
17. 扇区备用 低级格式化将一些块保留作为备用，对操作系统透明，当有坏块时，使用备用块来逻辑替换

5 输入输出

1. I/O设备分类
    - 按使用特性：人机交互（鼠标，屏幕，速度慢，以字节为单位） 存储设备（磁盘，速度块，以多字节组成的块为单位） 网络通信设备
    - 按传输速率：低速（字节） 中速（几千字节，打印机） 高速（数百千以上字节，光盘）
    - 按信息交换的单位：块设备（磁盘） 字符设备（打印机，终端机）
2. 程序控制方式
    - 程序直接控制 CPU需要不停检测与等待，串行工作
    - 中断驱动方式 每个字传输之后需要CPU确定
    - DMA（直接存储器存取）方式 DMA传输数据的整个过程不需要CPU参与，只需要在传输开始和结束才需要CPU （CR MAR MDR DC）
    - 通道 相当于一个专门处理I/O的CPU，是一种特殊的处理器
3. I/O子系统的层次结构 用户层I/O软件、设备独立性（无关性）软件、设备驱动程序、中断处理程序、硬件设备
4. 设备控制器 比如DMA方式就是使用了设备控制器，有数据线、信号线、控制线；数据寄存器和状态寄存器
5. 计算机系统为每台设备确定一个编号，这个编号称为设备的绝对号
6. 系统调用命令是用户操作的，与设备无关的，因此将参数翻译成设备操作命令的工作是由设备无关的操作系统完成的
7. 缓冲技术 T：I/O设备到缓冲区 M：缓冲区到内存 C：进程处理时间
    - 单缓冲 设备和处理机之间设置一个缓冲区 Max(C, T)+M
    - 双缓冲 Max(M+C, T)

8. 高速缓存和缓存区的区别 高速缓存是高速存储器，存放的是低速设备上某些数据的复制；缓存区设在内存中，设备和处理器之间
9. SPOOLing技术 以空间换时间，虚拟设备，只有一台打印机，并且正在被占用，但是依然可以发送打印任务，先把任务数据存放在缓存区中
    - 输入设备 -— 高速缓存（内存） -— 输入井（外存）
    - 输出设备 -— 高速缓存（内存） -— 输出井（外存）
    - 设备与输入井、输出井之间的数据传送由操作系统实现的，并非进程控制
    - 由预输入程序、井管理程序、缓输出程序管理
10. 设备分配
    - 设备控制表DCT 一个设备控制表就表征一个设备，里面保存设备的各项属性
    - 控制器控制表COCT 一个设备需要一个表项指向控制器，一一对应；该表表项是控制器的属性
    - 通道控制表CHCT 一个设备控制器请求对应的通道给它发命令，而一个通道可以给多个控制器服务，所以是多对一
    - 系统设备表SDT 每个设备占一个表项
    - SDT – DCT – COCT – CHCT
11. 设备分配的安全性
    - 安全分配算法 进程发出一次I/O请求之后就阻塞，不继续请求资源
    - 不安全分配算法 进程发出一次I/O之后还可以继续请求资源
12. 逻辑设备表LUT 将逻辑设备名映射为物理设备名
13. 设备分配方式 静态分配（独占设备的分配，提前分配好） 动态分配（）

Date: 2018-12-03

Author: hiro

Created: 2018-12-06 四 22:11