

期末复习

操作系统概念

- 操作系统的定义：
 - 操作系统是一组控制和管理计算机资源、对各类作业进行调度以及方便用户的程序的集合
- 操作系统的作用：
 - 用户与计算机硬件系统之间的接口
 - 计算机资源的管理者
 - 扩充机器
- OS设计的目标：方便性和有效性是最重要的目标
 - 方便性
 - 有效性
 - 开放性
 - 可扩充性
- OS的发展动力：
 - 不断提高计算机利用率的需求
 - 方便用户
 - 器件的不断更新换代
 - 计算机体系结构的不断发展
- 脱机I/O：
 - 目的：解决人机矛盾和CPU与I/O设备之间速度不匹配的矛盾
 - 方法：利用低速的外围机进行IO，例如：纸带（卡片）→磁带（磁盘）
- 2阶段调度：作业和进程，前者决定哪个（些）作业进入系统运行，后者决定为哪个进程分配CPU运行
- 操作系统类型：
 - 批处理
 - 分时
 - 实时
- 操作系统提供服务的最基本方式：系统调用
- 操作系统的特征：（并发和共享是操作系统最基本的特征）
 - 并发：
 - 并发是两个或多个事件在同一时间间隔内发生
 - 并行是两个或多个事件在同一时刻发生
 - 共享
 - 虚拟
 - 异步

- IPC通信的两个模型：消息传递和共享内存
- IPC的五种机制：信号、信号量、管道、消息队列、套接字
- 用户接口：命令、编程、图形
- 系统调用参数的传递方式：栈、寄存器、块或者表
- 内核的几种模式：
 - 无结构，或者说简单结构
 - 单一大内核
 - 模块化结构
 - 层次结构
 - 微内核
 - 混合结构
 - 外核
- 机制与策略相分离：机制决定如何去做（定时器），策略决定做什么（为每个用户决定将定时器设置为多长）

进程

- 前驱图：描述进程执行先后顺序
- 程序顺序执行：
 - 顺序性
 - 封闭性
 - 可再现性
- 程序并发执行：
 - 间断性
 - 失去封闭性
 - 不可再现性，为满足可再现，任意两个进程不能存在读写、写写冲突
- 进程：
 - 概念：进程是进程实体的运行过程，是系统进程资源分配和调度的一个独立单位
 - 性质：
 - 动态性：创建而产生，调度而执行，得不到资源而阻塞，撤销而终止（最基本的特征）
 - 并发性：不同进程可以并发进行
 - 独立性：进程是系统独立运行的一个单位，是资源分配和调度的独立单位
 - 异步性：进程按照各自独立、不可预测的速度向前推进
 - 结构特征：程序段、数据段、进程控制块（PCB是进程存在的唯一标志）
 - 三状态：就绪、执行、阻塞
 - 五状态：创建、就绪、执行、阻塞、终止
 - 七状态：创建、就绪、执行、阻塞、静止就绪、静止阻塞
- 几种调度区别：

- 长期CPU调度：选择进程进入就绪队列，发生地不频繁
- 短期CPU调度：选择应该执行的进程分配CPU，发生地非常频繁
- 中期CPU调度：对换，将部分程序从内存中调出，之后再调入内存执行
- 长期CPU调度控制着多道程序并发的程度，因为它控制了进程的数量
- 中期CPU调度影响进程数量，影响多道程序并发的程度
- 两种进程：
 - I/O密集型进程：主要进行I/O操作
 - CPU密集型进程：主要进行CPU操作
- 进程创建：
 - 子进程和父进程之间可能共享所有资源，可能部分，可能不共享
 - fork()，返回子进程的进程号，子进程自己fork()的号是0，子进程复制父进程的所有资源，若有exec()系统调用则将替换子进程的内容为新的程序
- 进程终止：
 - 自己终止：exit()
 - 其他进程终止
 - 用户终止：kill或者pkill
 - 父进程终止：父进程终止则子进程也终止
- 进程间通信（IPC）：
 - 消息传递机制：
 - 适合传递小数量信息，更容易实现，通过系统调用传递比如send()、receive()，在分布式系统中很有用
 - P、Q先建立交流连接，然后通过send()、receive()传递信息，交流连接可以是实体的也可以是逻辑的
 - 直接交流：
 - send(P,message)把消息发给P，receive(message,Q)收到了来自Q的信息
 - 对称和不对称：receive(id,message)收到来自任何人的信息
 - 不直接交流：
 - 通过邮箱mail交流，send(A,message)将消息发给邮箱A，receive(A,message)收到来自A的消息
 - 当两个进程有相同邮箱时可以建立联系，一个进程可以和多个进程联系，两个进程可能有多个不同的联系，联系可以是直接或者不直接的
 - 同一时刻只有一个进程可以收或者发信息，系统挑选进程
 - 邮箱归系统或者进程所有
 - 阻塞发送：发送者停止发送知道发送出的信息被收到；非阻塞发送：发送者一直发送信息。
 - 集合点：发送和接受都是阻塞的点。
 - 通过阻塞发送接收可以解决PC问题。
 - 共享内存机制：即生产者-消费者问题，有一个内存buffer

进程调度

- 进程调度情况，前三个需要调度，后两个必须执行
 - 从创建到就绪
 - 从等待到就绪
 - 从运行到就绪
 - 从运行到等待
 - 从运行到终止
- 调度形式：
 - 抢占式调度
 - 非抢占式调度
- 护航效应：其他进程等待一个长进程执行完毕（短进程等待长进程）
- 中断延迟时间：CPU收到中断到中断处理程序开始的时间。
- 调度延迟时间是调度程序进行进程切换花费时间，包括处理不同进程矛盾、切换上下文、切换用户模式、跳转到用户程序的合适位置等时间。
- 几个概念：
 - CPU使用率：CPU的利用率，尽可能大
 - 吞吐量：在一个时间单元内，CPU完成的进程数量，短进程吞吐量大，长进程吞吐量小
 - 周转时间：进程从提交到执行完毕的间隔，包括等待进入内存、在就绪队列等待、CPU上执行、I/O执行时间
 - 等待时间：进程在就绪队列中等待的时间。
 - 响应时间：针对交互系统而言，从用户产生需求（中断）到计算机响应的时间，而非输出结果的时间。包括中断延迟时间、调度延迟时间、进程实时执行时间。
- 调度算法：
 - FCFS算法
 - SJF算法
 - 优先级算法
 - RR算法
 - 多级队列：多个就绪队列，采用不同调度算法
 - 多级反馈队列：进程在不同调度算法队列之间迁移，从高到低
- 两种实时调度：
 - 硬实时调度：要求进程在截止时间之前完成
 - 软实时调度：要求关键进程比优先级更低的进程先执行
- 实时调度算法：
 - RM算法：周期越短优先级越高（固定），最优算法
 - EDF算法：截止时间越近优先级越高（动态）
 - 比例分配算法
- 优先级倒置：
 - 高优先级进程被低优先级进程延迟或者阻塞
 - 解决方法：

- 优先级继承：低优先级进程继承被其阻塞的高优先级进程，执行完释放阻塞资源让被阻塞进程执行
- 令阻塞高优先级的低优先级进程不可被抢占

线程

- 多线程编程的优点：
 - 响应性
 - 资源共享
 - 经济
 - 可伸缩性
- 多线程模型：
 - 一对一模型
 - 多对一模型
 - 多对多模型

进程同步

- 信号量

死锁

- 死锁的四个条件：
 - 互斥
 - 持有并等待
 - 非抢占
 - 循环等待
- 死锁的预防：从死锁的四个条件反向
- 死锁的避免：
 - 资源图算法，避免有环出现不安全状态
 - 银行家算法：三个矩阵：Allocation, Max, Need, Available 和安全检测算法
- 死锁检测：
 - 等待图：将资源图中的资源点去除
 - 银行家算法变种

- 死锁恢复：
 - 进程终止：
 - 终止所有死锁进程，代价很大
 - 一次终止一个进程，直到消除死锁，代价也很大，因为终止一个进程之后要调用算法检测死锁
 - 资源抢占：
 - 选择牺牲某些进程，释放资源
 - 回滚，将被抢占资源的进程回滚
 - 如何处理饥饿？

内存

- 存储层次结构：
 - 寄存器-Cache-内存-电子磁盘-磁盘-光带-磁带
 - 程序在内存中执行
- 内存的保护：每个进程分配基地址寄存器和界限寄存器，进程不可访问以外的地址，其他地址也不可能访问该进程地址，这两个寄存器的修改是特权
- 逻辑地址和物理地址：
 - 逻辑地址：每个地址在程序中的相对地址
 - 物理地址：每个地址在内存中的真正地址
- 地址绑定：在编译和装入阶段地址还是逻辑地址，在运行阶段通过MMU重定位寄存器绑定到物理地址
- 三种装入模式：
 - 绝对装入：在编译时期就是物理地址，适合小计算机系统和单道程序系统。
 - 可重定位装入：在装入时将程序中的指令和数据地址中的逻辑地址修改为物理地址，在装入时一次完成不再改变，也称为静态重定位。
 - 动态装入：实际清空，一个进程在内存中的位置可能多次变化，可能被换入换出，因此装入时并不把逻辑地址改为物理地址，在运行时才进行转换。
- 三种链接模式：
 - 静态链接：程序运行之前将各模块链接成一个完整的模块不再拆分。
 - 装入时链接：在程序装入内存时一个个链接，便于修改和更新，便于目标模块的共享。
 - 运行时动态链接：在程序执行时执行到某个需要的模块再链接。
- 单一连续分配：最简单的模式，内存简单划分为内核区（低地址区）和用户区（高地址区）
- 多分区分配：多道程序度依赖于分区的数量
 - 固定分区：每个分区大小相同
 - 动态分区：用双向链表链接分区
 - 首次适应
 - 循环首次适应
 - 最佳适应

- 最差适应
- 紧凑：将动态分配的不连续分区拼凑在一起，减少外部碎片，需要运行时重定位技术，并且涉及I/O问题，会带来延迟
- 两个碎片：
 - 外部碎片：在内存的分配和回收过程中，产生的空闲分区不连续，虽然总量足够，但是无法满足大块连续内存分配需求，主要发生在动态分区
 - 内部碎片：分配给进程的内存大于进程需求，多余内存无法得到利用，主要发生在固定分区
- 覆盖和对换：覆盖将某个进程执行需要的部分调入内存，不需要的部分放在磁盘，针对单个进程；对换则将执行的进程调入内存，某些不执行的进程调入磁盘，针对所有进程
- 分页式：
 - 页表，TLB（快表），EAT有效访问时间
 - 内存保护：页的有效位和无效位，是否在页表中
 - 多级页表：解决了页表要求大连续空间的问题，但是无法解决内存问题
 - 哈希页表
 - 反置页表

虚拟内存

- 虚拟存储器的特征：
 - 多次性
 - 轮换性
 - 虚拟性
- 按需调页：当出现缺页时将需要的页调入内存
- 写时复制：子进程修改的页面在写入修改时复制副本进行修改，其他未修改的页面和父进程共享
- 抖动：因为频繁调入调出页面，导致CPU利用率大大降低
- 工作集模型：进程在某段时间使用的页面集合

文件系统

- 文件属性：
 - 名称
 - 标识符
 - 类型
 - 位置
 - 尺寸
 - 保护

- 时间、日期和用户标识
- 文件操作：
 - 创建文件
 - 写文件
 - 读文件
 - 重新定位文件
 - 删除文件
 - 截断文件
- 打开文件表：维护所有打开文件的信息，请求文件时，可通过该表的索引指定文件，不需要搜索。对于多个进程可以同时打开文件的环境，采用两级内部表：每个进程表和整个系统表。包括以下内容：
 - 文件指针
 - 文件打开计数
 - 文件的磁盘位置
 - 访问权限
- 文件锁：
 - 读者锁和写者锁
 - 强制锁和建议锁，前者强制不可解，后者可视情况解
- 文件结构：
 - 逻辑结构：从用户出发看到的文件组织形式，又称为文件组织
 - 物理结构：系统将文件存储在外存的存储组织形式，用户无法看见
- 文件访问方法：
 - 顺序访问：按顺序一个个处理，可以读可以写，可以前进可以回退
 - 直接访问或者相对访问：允许程序按任意顺序进行读取和写入，基于磁盘模型，因为磁盘允许对任何文件块的随机访问。
 - 索引访问：建立在直接访问基础上，为每个文件块建立索引，通过索引快速查找文件块，为防止索引过大，可以为索引建立索引文件。
- 文件存储：
 - 通过磁盘分区，每个分区可创建文件系统，包含文件系统的分区称为卷，由目录和文件组成，目录由包含每个文件信息的结点组成
 - FCB：文件控制块，存储文件的基本信息
- 目录的结构：
 - 单级目录：所有文件都在一个目录里面
 - 两级目录：每个用户有自己的单级目录
 - 树形目录：绝对路径和相对路径，要删除只能递归
 - 无环图目录：允许共享子目录和文件，实现共享的方法
 - 软链接（符号链接）：是一种特殊类型的文件，包含一个文件名的路径名
 - 硬链接：允许一个文件在文件系统中有多多个文件名，都指向同一个索引结点
 - 通用图目录：可以有环
 - 垃圾收集机制

- 文件安装（挂载）：文件系统在用于系统的进程之前必须先安装，在安装点（挂载点）安装，即文件路径
- 文件共享：多用户共享和多系统共享
- 文件保护：
 - 访问类型：不同操作的限制
 - 访问控制：通过访问控制列表（ACL）规定每个用户的名称和允许访问的类型

文件系统实现

- 磁盘结构：
 - 引导控制块
 - 卷控制块
 - 目录结构
 - FCB
- 内存结构：
 - 安装表
 - 目录结构
 - 整个系统的打开文件表
 - 每个进程的打开文件表
- 虚拟文件系统（VFS）：操作系统中统一管理不同文件系统访问的抽象层，是物理文件系统和服
- 务的一个接口层，对不同文件进行抽象，使得不同文件系统在核心看来都是一样的
- 目录实现：
 - 线性列表
 - 哈希表
- 文件分配方法：
 - 连续分配：每个文件的文件块连续向后分配
 - 链接分配：
 - 隐式链接：目录标注文件的起始和终结结点，每个结点执行后一个结点，可以将多个块组成一个簇
 - 显式链接：
 - FAT（文件分配表）将用于链接文件各个物理块的指针显式地存储在一张链接表中，每个卷的开头部分用于存储该表
 - 可以对FAT采用缓存，减少磁头寻道时间
 - 索引分配：
 - 单级索引
 - 组合索引
- 空闲空间管理方法：
 - 位向量

- 链表
- 成组链接
- 计数
- 空间图
- 效率（空间）
- 性能（时间）：
 - 同步写和异步写
 - 随后释放和预先读取
- 恢复：
 - 一致性检查
 - 基于日志的文件系统

磁盘

- 磁盘的结构：
 - 磁臂
 - 柱面（确定磁臂所在的半径）
 - 扇区（确定磁臂所在的角度）
 - 盘片
- 磁盘的几个性能：
 - 旋转速度：每秒60-250转，即按照每分钟转数来计
 - 传输速率：在驱动器和计算机之间数据流的速率
 - 定位时间（随机访问时间）：包括寻道时间（将磁臂移动到所要柱面的时间）和旋转延迟（旋转磁臂到所需扇区的时间）
 - 磁盘带宽：传输字节的总数除以从服务请求开始到最后传递结束时的总时间
 - 磁盘是可移动的
- 磁头碰撞：磁头损坏磁盘表面
- 调度算法：
 - FCFS：先到先处理
 - SSTF：选择距离当前最近的处理
 - SCAN（电梯算法）：磁头从一端移动到另一端，然后又从另一端移动到另一端
 - C-SCAN：移动到另一端之后就立即返回另一端，不处理任何请求
 - LOOK：和SCAN一样，但是不移动到最远，只移动到请求的最远
 - C-LOOK：和C-SCAN一样，区别与LOOK和SCAN区别一样
- 磁盘管理：
 - 磁盘格式化：
 - 低级格式化或者物理格式化：将磁盘分为可以读写的扇区
 - 逻辑格式化或者创建文件系统：将初始的文件系统数据结构存储到磁盘上

- 引导块：自举程序处存储在只读存储器ROM中，完整的引导程序在磁盘固定位置上的启动块，具有启动分区的磁盘称为启动磁盘或者系统磁盘。
- 坏块：损坏的磁盘块，可以通过扇区备用（备用的磁盘块）替代或者扇区滑动（将坏块向后移动，重新映射所有对应块）
- 对换空间管理
- RAID：磁盘冗余阵列，提高性能和可靠性，原来目标是廉价现在是独立
 - RAID0：无冗余，所有盘都是数据盘，采用块级分条，文件的块分布在N个盘上
 - RAID1：每个磁盘都有一个镜像，一个逻辑磁盘由两个物理磁盘构成（镜像卷），每次写入两个磁盘。
 - RAID2：采用位级分条和ECC纠错（海明码纠错）
 - RAID3：采用位级分条和奇偶校验位，只需要一个奇偶校验盘
 - RAID4：采用块级分条和奇偶校验位
 - RAID5：采用块交错分布式奇偶校验
 - RAID6：采用P+Q冗余方案
 - RAID0+1：先分条再镜像
 - RAID1+0：先镜像再分条
 - RAID0、RAID1、RAID5、RAID1+0

I/O

- I/O控制机制：
 - 轮询
 - 中断
 - DMA（直接内存访问）
 - I/O通道
- 周期窃取：当DMA控制器占用内存总线时，CPU被暂时阻止访问内存，但是仍然可以访问主缓存或者辅助缓存中的数据项
- 设备独立性：应用程序和具体的物理设备无关
- 阻塞I/O：进程被阻塞直到I/O完成
- 非阻塞I/O：进程在I/O执行时也运行
- 缓冲Buffer的理由：
 - 处理数据流的生产者和消费者之间的速度不匹配
 - 协调传输大小不一数据的设备
 - 支持应用程序I/O的复制语义
- 缓冲机制：
 - 单缓冲
 - 双缓冲
 - 循环缓冲

- 缓冲池
- 脱机和假脱机：
 - 脱机：脱离网络联机，没有连接到计算机或者其他设备的情况下进行操作
 - 假脱机：外部设备联机并行操作