# 第2章 进程

## 进程是啥

1. 进程是**过程**，是一个程序在一个数据集合上的一次运行的过程。
2. 进程的特征：动态性(由定义决定的)； 并发性(一段时间内多个进程交替运行)

独立性(分配资源的基本单位)； 异步性(不同进程间的相互作用，执行结果不可复现)；

结构性(进程实体由程序段、数据段和PCB组成)

1. 创建、就绪、运行、等待、结束。常考三状态转换和进程的创建。
2. PCB中包含进程描述信息(PID、UID)、进程控制管理信息(当前状态、优先级、程序地址、进入内存时间、CPU占用时间等，页表基址和长度maybe)、资源分配清单(代码段数据段指针、文件描述符、IO设备等)、处理机相关信息(各寄存器值)
3. PCB常驻内存，是进程存在的唯一标志。各进程的PCB通常用链接或索引组织在一起，相同状态的进程的PCB放在同一个队列或索引表中。
4. 进程间通信：共享存储，进程间划一块可直接访问的区域，需要同步互斥工具来管理；消息传递，直接、间接；管道(共享存储的升级优化)，连接两个进程的一个共享文件(缓冲区)，固定大小，一头写一头读，写进程一次写满，读进程才能读，一次读完就清空。
5. 线程：轻量级进程。处理器调度的基本单位。有各自的唯一标识符和线程控制块。不拥有资源，只保持各自的栈指针不共享。切换时不用保存和重置CPU环境，开销小。
6. 线程实现方式：用户级多线程、内核级多线程。模型：多对一、一对一、多对多。

## 进程调度

1. 三级调度，作业调度：从外存上处于后备状态的作业中挑选出来，分配内存、IO资源，并建立进程，使其获得竞争处理器的权利。对于每个作业只调入一次调出一次。中级调度：就是内存的交换技术。进程调度：按一定原则给进程分配处理器资源。
2. 三种调度的频率是递增的。进程调度是最基本的。作业调度可看作是进程活动的准备工作。
3. 作业与进程的区别：作业是用户提交的，以用户任务为单位。进程是系统自动生成的，以操作系统控制为单位。
4. 禁止进程切换的情况：处理中断的过程中、进程进入**操作系统内核程序**的临界区(不是所有的临界区都符合)、屏蔽中断的原子操作过程中。
5. 调度评价指标，CPU利用率、系统吞吐率(单位时间内完成作业的数量)、周转时间(作业提交到完成的时间)、带权周转时间(占用比，周转时间/实际运行时间)、响应时间(提交到首次响应间等待的时间)、等待时间(总的)
6. 典型调度算法：先来先服务(FCFS)，短作业优先(SJF)，优先级(静态、动态)，时间片轮转，高响应比(主要用于作业调度，)，多级反馈队列
7. 典型考点：计算某一算法下，一批作业的周转时间，某一时刻选中哪个作业。
8. 考察各算法特点。比如高响应比和时间片不会产生饥饿现象。**I/O型作业**优先级高于计算型作业。时间片适用于分时系统。高响应比综合考虑等待时间和执行时间。
9. 一类题型：考察多道作业批处理。每道作业都有相同的几步工序。不同的工序使用不同的设备资源并行工作。合理安排每道作业的工序，使总的完成时间最短。画甘特图最保险，横坐标为时间，纵坐标为不同进程，在每段线段上标注进程所用的设备或CPU。

## 进程同步

1. 临界资源：一次只能为一个进程使用的资源。临界区：访问临界资源的那段代码。
2. 软件实现同步互斥：Peterson算法
3. 硬件实现：TestAndSet指令，Swap指令，不符合让权等待。
4. 信号量实现：P操作，V操作

经典问题：生产者-消费者，读者-写者，哲学家进餐

1. 隐藏条件，信箱、货架都当作缓冲区，必须互斥访问。P98.22
2. 读者-写者类问题中的同类用if加塞。P97.18
3. 灵活使用if…else…结构。P97.17
4. 保证互斥的前提下最大限度并行，用设立多个信号量做到。P98.23
5. 管程：PV操作大量分散在各进程中，不易管理，引入管程封装同步互斥不易出错。1）局部于管程的共享结构数据的说明。2）该数据结构的一系列操作。3）对局部于管程的共享数据设置初始值的语句。

## 死锁

1. 定义：多个进程因竞争资源而陷入僵局(相互等待)。产生原因：系统资源竞争、进程推进顺序非法
2. 死锁产生的4个必要条件，不满足一个就不会产生死锁。全满足不一定死锁(圈外人有同类资源)

互斥：资源使用的排他性。不剥夺：进程获得的资源在其使用完毕前不能被剥夺。

请求和保持：脚踏几条船还坚持继续撩骚。循环等待：狗咬狗形成环。

1. 死锁预防：从3个必要条件(互斥条件是不能动的)下手。
2. 破坏不剥夺：阻塞的进程必须放弃已占有的资源(和被抢没差)。
3. 破坏请求和保持：必须一次申请到所有要的资源后才可以开始运行。开始运行后已占资源可以持续拥有，但不允许申请新的资源。
4. 破坏循环等待：所有资源编号，进程只准按编号顺序申请资源。
5. 死锁避免：银行家算法，一个向量Available，三个矩阵Max = Allocation – Need，一旦找不到执行的安全序列就拒绝该进程的申请。
6. 死锁的检测：资源分配图，找出未阻塞的进程，消除他和他的请求边和分配边。若无法全部消除，则存在死锁。
7. 死锁的解除：资源剥夺法、撤销进程法、进程回退法

# 第3章 内存

## 内存分配

1. 编译🡪链接🡪装入🡪重定位

编译：将高级语言翻译成机器语言，并形成若干模块，各模块逻辑地址都从0开始。

链接：将各模块和库函数连起来，形成完整装入模块，各模块逻辑地址统一。

将装入模块放到内存里之后，需要修改目标程序中指令和数据使用的逻辑地址，改为实际的物理地址，这个过程称为重定位。

动态链接与分段管理联用。猜测动态运行时装入一般和请求分页管理联用，需要某页，装入某页，再重定位某页。

1. 覆盖：程序员从系统分配给某进程的分区中划出固定区和覆盖区，覆盖区内可换入换出

交换：配合进程调度，将等待状态的进程占的内存换出到辅存，将就绪进程的换入内存

进程间的换入换出，和虚存同思路，但不完全一样，可能是虚存的前身。

1. 连续分配，单一连续分配；固定分区分配：分区大小相等/不等；动态分区分配：首次适应、循环首次适应、最佳适应、最坏适应。最佳适应反而不如首次适应。

易忽视点：空闲分区合并时前后兼顾

1. 非连续分配，分页式、分段式、段页式。抓住三个重点：逻辑地址结构、表项结构、寻址过程。特别注意寻址过程的界地址保护。
2. 段页式的寻址过程与多级分页有所不同，具体思考在P156底P157图注释
3. 逻辑地址和物理地址转换，特别是16进制下的计算P163.4
4. 页面大小选取的优劣，页面大页表就小，但进程最后一页内部碎片大。反之同理。
5. 页表项大小选取的计算规则。页表大小的计算
6. 顶级页表只能有一个页面。非顶级一般也只有一页P190.29、P162.51
7. 每个进程都有一张页表。大多数进程的页表都驻留在内存中。系统中只设置一个页表寄存器PTR，在其中存放页表起始地址和页表长度。当进程未执行时，页表起始地址和长度存放在PCB中。

## 虚拟内存

1. 是什么：以逻辑地址与物理地址转换、非连续内存分配、局部性理论、类似交换技术为前提，从逻辑上扩展内存容量

虚存的容量也就是逻辑上内存容量，无关实际主存辅存大小，地址有几位就能有多大

1. 请求分页/段管理 = 分页/段管理 + 虚拟内存 + 缺页中断机制

不用把程序一次性全部装入内存，只用装入起步部分就行，后面有需要就中断调页

1. 页面置换策略：固定分配、可变分配；局部置换、全局置换

题型：考概念，固定分配 + 全局置换(大雾)

1. 抖动：某页刚换入就换出，刚换出又要换入，过度频繁缺页。原因是该进程常用的页面数量大于其被系统分配的页帧数量
2. 工作集：某段时间内，进程要访问的页面集合。同一页只记一次

题型：规定一个工作窗口大小，求某时刻的工作集

1. 局部置换常用的算法：OPT，FIFO，LRU，CLOCK(NRU)。Belady现象。

题型：给定一个页面访问序列，求缺页率/某时刻淘汰页，主要考察LRU和改进型CLOCK；

1. 计算访问时间难点：无TLB的情况P192.3，缺页用时为1次访存查页表+1次磁盘读写+1次访存查页表+1次访存拿数据。有TLB的情况P192.5，缺页用时为1次查TLB+1次访存查页表+1次磁盘读写+1次查TLB+1次访存拿数据。
2. CLOCK算法的一个难点：某一时刻指针所在位置是最新装入内存的那一页的下一页(题目出得严谨时)，或者是访问位刚归零的下一页(题目出得不严谨)
3. 综合题，地址翻译操作系统P184，计组P126.7和P138.1，与计组Cache章节联动。第一步将虚拟地址划分为虚拟页号+页内偏移。如果TLB采用直接映射或组相联，再将虚页号划分为TLB标记+TLB索引。检索TLB和页表，拿到完整的物理地址后，再划分为主存字块标记+Cache字块地址+块内偏移。

# 第4章 文件

## 文件逻辑结构

1. 用户首次打开一个文件时，系统需要将该文件的FCB拷贝到内存中的打开文件表中作为一个条目，并将一个指向该条目的指针(文件描述符)返回给用户。文件打开之后可以使用文件描述符进行读写操作。
2. 文件控制块FCB是用来存放文件的控制信息的数据结构。FCB的有序集合就是文件目录，文件目录中每个FCB成为文件目录项。
3. UNIX系统简化了文件目录，不直接将FCB作为文件目录项。UNIX的文件目录中仅有14字节的文件名和2字节的inode指针。索引结点inode中存放文件控制信息。这样就将原先目录中64字节的FCB缩减为16字节，提高了检索速率。
4. 基于索引结点实现文件共享，即硬链接，是指两个用户的文件目录中都有一个表项的inode指针指向某个共享文件的索引结点。索引结点中需要维护一个链接计数。当计数不为1时，某个用户无法删除该文件，只能删除自己的文件目录里的目录项，再将链接计数减1。
5. 基于符号链实现文件共享，即软连接，用户乙要共享用户甲的一个文件F，乙在自己的文件目录里创建一个文件也叫F，但新F中存放的是到原F的路径名。甲把原F删了，乙也就访问失败了。
6. 访问控制列表(存取控制矩阵)用来规定多个不同用户所允许的访问类型。

## 文件系统实现

1. 用户接口 🡪 文件目录系统 🡪 存取控制模块 🡪 逻辑文件系统 🡪 物理文件系统 🡪 设备管理模块 🡪 设备
2. 索引结点的结构及作用，文件的索引结点采用混合索引(有直接索引块，一级索引块，二级索引块…)，求单个文件最大长度，或求找到第k个字节要读盘几次
3. 空闲盘块的位示图法，分配/释放第m块盘块，修改哪一位
4. 熟悉文件目录的结构，熟悉FAT表的结构。FAT全磁盘仅一张，存入内存以减少读写次数。
5. 目录文件组织成链式文件，普通文件组织成索引文件，给出目录树，从根目录找到某个文件需要读盘几次？建议草稿上记录进入内存的目录文件内容而非目录项，更不要把两者搞混。

## 磁盘管理

1. 磁盘的平均存取时间Ta = Ts + Tr + Tt ，Ts寻道延迟，Tr旋转延迟，Tt传输延迟
2. 磁盘调度算法，先来先服务FCFS，最短寻找时间优先SSTF，电梯SCAN，LOOK(到达最远端请求即可返回的SCAN)，循环扫描C-SCAN，C-LOOK(到达最远端请求即可返回的C-SCAN)
3. **低级格式化**即物理分区将磁盘划分为扇区，每个扇区由头部、数据区(512B)、尾部组成，这样磁盘控制器才能进行读写。然后，操作系统再将磁盘分区(C盘、D盘)。最后，进行**逻辑格式化**，操作系统将初始的文件系统数据结构存到磁盘上，包括空闲空间、已分配空间、空的文件目录。
4. 计算机启动时需要运行自举程序，自举程序通常存在主存的ROM中。但为了避免修改自举代码就要改变ROM硬件的问题，现在ROM中只保留很小部分的自举程序，完整的部分保存在磁盘启动块上，启动块位于磁盘的固定位。拥有启动分区的磁盘即时系统盘。

# 第5章 I/O

1. 自陷、访管、特权指令三者的关系：

原则上等同于访管指令，但是从操作系统的角度叫的。访管强调的是cpu切换到了核心态，可以执行指令集中的所有指令。而陷入（自陷、陷阱）指令强调程序从用户程序切换到了操作系统，陷入指令即汇编中的中断指令，执行陷入指令程序中断，跳转到中断服务程序（操作系统的代码）。所有访管强调可以执行特权指令，陷入强调“跳转”到操作系统。核心态和操作系统是不可分割的：其实道理很显然，进入核心态（管态），必然需要“跳转”到操作系统，不然区分用户态和核心态就没有了意义，如果程序执行完访管指令后，只是进入了管态，之后仍然执行自身的代码，那用户程序将可以为所欲为，显然不能允许。另一方面，如果cpu跳转到了操作系统，没有进入管态，那操作系统也无法执行特权指令，管理整个机器，显然也不行。所有管态就是运行操作，访管指令本质上就是一条陷入（中断）指令。

https://blog.csdn.net/dollarser/article/details/102908079

1. I/O控制方式，见计组笔记。
2. I/O子系统的层次结构自上而下是：用户层I/O软件，设备独立软件，设备驱动程序，中断处理程序，硬件。用户层I/O软件实现与用户交互的接口。设备独立性软件实现用户程序和设备驱动器的统一接口。设备驱动器将系统调用参数翻译成设备操作命令，负责具体实现系统对设备发出的操作命令。驱动I/O设备工作。中断处理程序处理中断相关事项。硬件设备包括设备本身和控制器。具体举例参见操作系统P270
3. 在设备和处理器之间引入缓冲区，缓和速度不匹配，减少中断频率，解决基本数据单元不匹配，提高CPU和外设并行性。分类有单缓冲、双缓冲、循环缓冲、缓冲池。假定磁盘把一块数据输入到缓冲区时间为T，该缓冲区数据送到用户区时间为M，处理数据时间为C，单缓冲总用时max{C, T}+M，双缓冲总用时max{C+M, T}
4. SPOOLing技术：目的是为了将独占设备改造成共享设备。在磁盘上开辟出两个存储区域，输入井收容I/O设备输入的数据，输出井收容I/O设备输出的数据。脱机输出的过程：当用户有数据要输出，将主存上的数据送到磁盘上的输出井，此时对于上层的用户来说已经完成了输出。然后，等到外设空出来时，输出进程将输出井中的数据经过缓冲区送入输出设备真正完成输出任务。输入任务同理。

输入设备 🡪 输入缓冲 🡪 输入井 🡪 内存(用户)

内存(用户) 🡪 输出井 🡪 输出缓冲 🡪 输出设备

1. 设备分配要考虑I/O设备固有属性、I/O设备的分配算法、I/O设备分配的安全性、I/O设备的独立性。设备独立性是指应用程序独立于具体使用的物理设备。在应用程序中使用逻辑设备名称来请求使用某类设备。系统通过LUT逻辑设备表将逻辑设备名映射为物理设备名。
2. 设备分配的数据结构：系统设备表SDT，设备控制表DCT，控制器控制表COCT，通道控制表CHCT。整个系统只有一张SDT。每个设备一张DCT，DCT其中一个表项是指向COCT的指针。每个COCT有一个表项存放指向相应的通道的CHCT。CHCT中有一个或多个指针指向通道负责的设备的COCT。