区别

- **实地址和虚拟地址：实地址**：指的是主存中的地址，实际的主存储器的地址，对应主存空间，即物理空间。
虚地址：指的是存在于虚拟内存中的地址，它有时候在磁盘中，有时候在主存中。
- **API和系统调用的区别**：API 侧重于向上层提供服务，而**系统调用**则侧重于通过软中断向下层的内核发出一个明确的请求
- **多处理器系统和多核系统：多处理器系统**：包含两台或台功能相近的处理器，处理器之间彼此可以交换数据，所有处理器共享内存，I/O 设备，控制器，及外部设备，整个硬件系统由统一的操作系统控制，在处理器和程序之间实现作业、任务、程序、数组极其元素各级的全面并行。**多内核**是指在一枚处理器中集成两个或多个完整的计算引擎（内核）。
- **抢占式和非抢占式调度：非抢占**：在这种情况下，一旦进程处于运行状态，就会不断执行直到终止，进程要么因为等待 I/O，要么因为请求某些操作系统服务而阻塞自己。**抢占**：当前正运行进程可能被操作系统中断，并转换为就绪态。一个新进程到达时，或中断发生后把一个阻塞态进程置为就绪态时，或由现周期性的时间中断时，需要进行抢占决策。
- **竞争进程和合作进程：竞争进程**需要同时访问相同的资源像磁盘，文件或打印机。**合作进程**要么共享访问一个共有的资源，像一个内存访问区，要么就与其他进程相互通信，在一些应用程序或活动上进行合作。
- **强信号量和弱信号量**：强信号量要求在信号量上等待的进程按照先进先出的规则从队列中移出。弱信号量没有此规则
- **死锁避免、检测和预防**:处理死锁通常有三种方法：预防检测和避免。**死锁预防**通过确保不满足死锁的一个必要条件来避免发生死锁。操作系统总是同意资源请求时，需要进行**死锁检测**。操作系统必须周期性地检查死锁，并采取行动打破死锁。**死锁避免**涉及分析新的资源请求，以确定它是否会导致死锁，且仅当不可能发生死锁时才同意该请求。
- **两种局部性**: **时间局限性：**指程序中的某条指令或某个存储单元一旦被执行或访问，则在不久的将来可能会再次发生（因为程序中存在着大量的循环操作）；**空间局限性**:指一旦程序访问了某个存储单元，则不久的将来该存储单元附近的存储单元也最有可能被访问（因为程序是顺序执行的）
- **逻辑地址、相对地址和物理地址**. **逻辑地址**是指与当前数据在内存中的物理分配地址无关的访问地址，在执行对内存的访问之前必须把它转换为物理地址。**相对地址**是逻辑地址的一个特例，它是相对于某些已知点（通常是程序的开始处的存储单元。**物理地址**或**绝对地址**是数据在内存中的实际位置。
- **页和页框**. **页**：一页指一系列的线性地址和包含于其中的数据。**页框**：分页单元认为所有的 RAM 被分成了固定长度的页框每个页框可以包含一页，也就是说一个页框的长度和一个页的长度是一样的。页框是内存的一部分，是一个实际的存储区域。页只是一组数据块，可以存放在任何页框中。也就是说页对应的是线性（虚拟）地址的东西，而页框对应的是物理地址，是实际的存储区域。
- **页和段**.页大小相等，段可以大小不等；分页对程序员来说是透明的，而分段通常是可见的。
- **简单分页与虚拟内存分页**.进程运行时，**简单分页**的所有页必须都在内存中，除非使用了覆盖技术，**虚存分页**并非所有页都须在内存页框中，仅在需要时才读入页，把一页读入内存可能需要把另一页写出到磁盘。
- **分段与分页的区别**：页是信息的物理单位，段是信息的逻辑单位；分页式存储管理的作业地址空间是一维的，而分段式存储管理的作业地址空间是二维的；页的大小固定且由系统确定是等长的，而段的长度不定；分页的优点体现在内存空间的管理上，二分段的优点体现在地址空间的管理上。
- **陷阱与中断的主要区别**：**陷阱**可以嵌套发生，而**中断**默认情况下不可以嵌套；**陷阱**的中断源头、服务对象来自CPU的内部(错误、异常事件如除数为0，校对错误)，而**中断**的中断源头来自CPU外部,其它进程请求.外部事件)。
- **原语和系统调用的区别：操作系统层次不同**.原语属于操作系统，在操作系统面和硬件面之间；系统调用属于操作系统，在用户程序面和操作系统面之间。**术语概念不同**．原语是指一系列原子性的操作/针对硬件的一系列操作，一般的实现是通过指令实现的，比如开关中断，主要强调的是原子性也就是原子操作。系统调用强调的是一个接口性的问题，是用户程序和操作系统的接口，是为了获取系统服务而为用户程序提供的接口。系统调用强调在核心态执行的OS的程序，原语强调是操作系统核心态下程序可以调用的小程序，可以直接当成是命令，指令来理解。也可以用大程序和小程序简单理解。原语常驻内存，但是系统调用不是。
- **驻留集和工作集：驻留集**表示进程在内存中的页集，**工作集**表示进程在过去的一段时间中被访问到的页集。
- **缺页中断和越界中断：**缺页中断:在利用逻辑地址计算物理地址时，在得出的页号合法，却不在页表内时，由操作系统引发的中断；越界中断：和缺页中断类似，在判断合法时，若为非法，则发生越界中断
- **用户级线程较内核级线程的优点**：线程的调度不需要内核直接参与，控制简单。可以在不支持线程的操作系统中实现同一进程中只能同时有一个线程在运行，如果有一个线程使用了系统调用而阻塞，那么整个进程都会被挂起，可以节约更多的系统资源。**缺点**：一个用户级线程的阻塞将会引起整个进程的阻塞。用户级线程不能利用系统的多重处理，仅有一个用户级线程可以被执行
- **用户模式和内核模式：**用户模式下可以执行的指令和访问的内存区域都受到限制。这是为了防止操作系统受到破坏或者修改。而在内核模式下则没有这些限制，从而使它能够完成其功能。
- **驻留集管理和页面置换策略**：驻留集管理的概念为（1）给每个活动进程分配多少页框。（2）计划置换的页集是局限于那些缺页中断的进程，还是局限于所有页框都在内存中的进程。置换策略的概念为：在计划置换的页集中，
选择换出哪一页。
- **谨慎写和延迟写**：谨慎写是对写入操作进行逐个排序的写入方式；延迟写是把对文件的修改写入高速缓存中，然后在适当的时候选最佳方式把高速缓存内容刷新到磁盘上。

一些定义

- **用户态到内核态切换的三种方式：**系统调用、异常 、外设中断
- **交换：**把主存中某个进程的一部分或者全部内容转移到磁盘。当主存中没有处于就绪态的进程时，操作系统就把一个阻塞的进程换出到磁盘中的挂起队列，从而使另一个进程可以进入主存执行。
- **操作系统的内核**：操作系统内核是计算机上最低层的软件，提供计算机最核心的功能，如：进程管理、内存管理、I/O管理、文件管理、网络管理等。
- **计算机的四个组成部分：**处理器：控制计算机的操作，执行数据处理功能。内存：也叫主存储器，存储数据和程序。输入/输出模块：在计算机和外部环境之间移动数据。
- **处理器寄存器的两种类别：**用户可见寄存器、控制和状态寄存器
- **处理多个中断：**正在处理一个中断时，禁止再发生中断；定义中断优先级
- **被一个进程中的所有线程共享**：该进程的状态和资源，例如地址空间，文件资源，执行特权等。（不共享什么呢？）
- **两种使用接口：**用户接口和程序接口
系统调用时用户程序和操作系统之间的接口。
- **两种进程：**系统进程和用户进程。
- **机器指令能指定的四种不同操作：**处理器 . 寄存器：数据可以从处理器传送到存储器，或者从存储器传送到处理器；处理器 . I/O：通过处理器和 I/O 模块间的数据传送，数据可以输出到外部设备，或者从外部设备输入数据；数据处理：处理器可以执行很多关于数据的算术操作或者逻辑操作；控制：某些指令可以改变执行顺序
- **有信息表需要维护的实体**：内存、I/O、文件 和 进程。
- **进程控制块中的三类信息：**进程标识，处理器状态信息，进程控制信息。
- **创建一个进程的事件：**新的批处理作业；交互登录；操作系统因为提供一项服务而创建；由现有的进程派生。
- **系统设计的三个目标**：方便、 有效、 扩展的能力
- **并发的难点**：全局资源的共享充满了危险：. OS 很难对资源进行最优化分配；定位程序设计错误非常困难
- **系统总线**：在处理器、内存和输入输出间提供通信的设施。
- **高速缓存**：高速缓冲存储器是比主存小而快的存储器，用以协调主存跟处理器，作为最近储存地址的缓冲区。
- **中断**：计算机运行过程中，出现某些意外情况需主机干预时，机器能自动停止正在运行的程序并转入处理新情况的程序，处理完毕后又返回原被暂停的程序继续运行。
- **两种处理器寄存器**：用户可见寄存器.优先使用这些寄存器，可以使机器语言或者汇编语言的程序员减少对主存储器的访问次数。对高级语言而言，由优化编译器负责决定把哪些变量应该分配给主存储器，一些高级语言，如 C 语言，允许程序员建议编译器把哪些变量保存在寄存器中。控制和状态寄存器：用以控制处理器的操作且主要被具有特权的操作系统例程使用，以控制程序的执行。
- **互斥要求**.必须强制实施互斥：在与相同资源或共享对象的临界区有关的所有进程中，一次只允许一个进程进入临界区。一个在非临界区停止的进程不能干涉其他进程。绝不允许出现需要访问临界区的进程被无限延迟的情况，即不会死锁或饥饿。没有进程在临界区中时，任何需要进入临界区的进程必须能够立即进入。对相关进程的执行速度和处理器的数量没有任何要求和限制。一个进程驻留在临界区中的时间必须是有限的。
- **产生并发的三种上下文：**多应用程序，结构化应用程序 ，操作系统结构
- **通道：**又称I/O处理器，是独立于CPU的专管输入输出控制的处理器，控制设备与内存直接进行数据交换。
- **进程上下文：**执行上下文又称为进程状态，是操作系统用来管理控制所需的内部数据。这种内部信息和进程是分开的，因为操作系统信息不允许被进程直接访问。上下文包括操作系统管理进程以及处理器正确执行进程所需要的所有信息，包括各种处理器寄存器的内容，如程序计数器和数据寄存器。它还包括操作系统使用的信息如进程优先级以及进程是否在等待特定 I/O 事件的完成。
- **可重用资源的例子：**处理器 、I/O 通道 、内存和外存 、设备、文件、数据库和信号量之类的数据结构。
- **进程状态：**新建态、就绪态、 运行态、 阻塞态、 挂起态、退出态 。**进程映像**：程序 。数据 。栈 。属性。**进程标识信息**：存储在 PCB 中的数字标识符，包括：进程 ID，父进程 ID，用户 ID。**进程控制块中的信息**：进程标识，处理器状态信息，进程控制信息。进程总具有**对资源的控制权或所有权**，这些资源包括内存、I/O通道、I/O设备和文件等。
- **内存管理满足的需求：** 重定位 保护 共享 逻辑组织 物理组织。
- **页表项中能找到的典型元素**：页号 进程标志符 控制位 链指针
- **记录组块**：定长组块、变长跨越式组块 、变长非跨越式组块
- **文件组织**：顺序文件：每条记录都使用固定的格式，所有记录具有相同的长度，并由相同数量、长度固定的域按特定顺序组成。每条记录的第一个域通常是关键域；索引顺序文件：用于支持随机访问的文件索引和溢出文件。索引顺序文件极大的减少了访问单条记录的时间；堆：是简单的文件组织形式，数据按它们到达的顺序被收集
- **操作系统的五种典型存储管理职责**：进程隔离 、自动分配和管理、支持模块化程序设计 、保护和访问控制、 长期存储
- **管程**：由一个或多个过程，一个初始化序列和局部数据组成的软件模块，其主要特点如下：局部数据变量只能被管程的过程访问，任何外部过程都不能访问。一个进程通过调用管程的一个过程进入管程。在任何时候，只能有一个进程在管程中执行，调用管程的任何其他进程都被阻塞，以等待管程可用。（用于实现进程的互斥，还可以用于实现其他进程间的同步操作，例如条件变量的等待与唤醒。）
- **内部碎片：**分区中已分配给某些作业，但未被使用的空闲部分.
- **外部碎片：**经过一段时间运行，内存中有多个因为太小而无法分配给其他作业使用的小分区。
- **文件的操作：**文件创建、文件删除、文件截断、文件读、写、读写定位、打开、关闭。
- **文件目录：**将文件名与外存空间的物理地址建立对应关系，体现这种关系的数据结构称为文件目录。
- **目录文件：**把若干文件目录组织在一起，以文件的形式保存在外存上，以备后用，形成了目录文件。

- **进程调度算法：**先来先服务(FCFS)、时间片轮转调度高响应比优先(非抢，响应比=（t等待+t服务）/t服务）短进程优先法(SFJ，非抢)、优先级法、多级反馈队列轮转法、最短剩余时间优先法（抢占式）
- 周转时间=完成时间-到达时间
带权周转时间=周转时间/运行时间（服务时间）
平均…=…/进程数
等待时间=周转时间-运行时间=进程开始执行的时间-到达时间
- 长程调度=作业调度=高级调度
中程调度=内存调度=中级调度
短程调度=进程调度=低级调度
长程调度：决定加入待执行进程池。
中程调度：决定加入部分或全部位于内存中的进程集合
短程调度： 决定可用 I/O 设备处理哪个进程挂起的 I/O 请求。

- **（内存）动态分区分配算法**：首次适应法（首个能装下的）、邻近适应法（从上次分配内存分区的下一个分区开始查找）、最佳适应法（大小最相近）、最大（坏）适应法
- **磁盘调度算法：**电梯调度（SCAN），循环电梯调度（C-SCAN），先来(FCFS)，最短寻道（SSTF）
RAID:独立磁盘冗余阵列
- **磁盘高速缓存的置换策略：最近最少使用（LRU）**：置换在高速缓存中未被访问时间最长的块，可用一个栈完成。
最不常使用（LFU）：置换集合中访问次数最少的块。

访存次数：
页式存储：（2次）
第一次，访问内存中的页表，利用逻辑地址中的页号查找到物理块号，与逻辑地址中的页内偏移拼接形成物理地址；
第二次：得到物理地址后，再一次访问内存，存取指令或者数据。
段式存储：2次（同上）
段页式存储：（3次）
第一次：访问内存中的段表，查到页表的起始地址
第二次：访问内存中的页表，找到页帧号，形成物理地址
第三次：得到物理地址后，再一次访问内存，存取指令或者数据
多级页表
若页表划分为N级，则需要访问内存N+1次。
若系统有快表，则在快表命中时，只需访问1次内存即可。
引入快表：
因为把页表放在内存中，至少需要访问两次内存才能存取一条指令或者数据（一次得到物理地址地址，一次存取），比较慢；
为此在地址变换机构中增设了一个具有并行查找能力的高速缓冲寄存器.快表（全局只有一个，不在内存中！！！）
用来存放当前访问的若干页表项（比较小，只能存放部分页表项）
快表命中
可直接得到物理块号，与页内偏移拼接成物理地址后访问内存，进行指令或者数据的存取。（只需访问一次内存）
快表不命中
第一次，需去内存中访问页表，形成物理地址
第二次，根据物理地址，访问内存进行指令或者数据的存取（需要访问两次内存）

段页式管理机制就是把主存先分成若干段，每个段又分成若干页，也就是说段页式管理机制中段与段之间以及段的内部的都是离散的。分段有外部碎片，分页有内部碎片；一个进程只有只有一个段表，但是可以有多个页表。

特性

- **多道程序设计技术**：在计算机内存中同时存放几道相互独立的程序，使他们在管理程序控制之下，相互交替运行，使CPU及外部设备尽量处于忙碌状态。**特征**：多道、宏观上并行、微观上串行。
- **分时系统**：把处理器运行时间划分成很短的时间片，轮流的分配给各联机作业使用，如果某个作业在分配给它的时间片用完之前还未完成，改作业就暂时中断，等待下一轮继续计算，此时处理器让给另一个作业使用。**特点**：多路性、独占性、交互性、及时性。
- **实时系统**：及时响应、高可靠性和安全性、系统的整体性强、交互回话功能较弱。
- **操作系统**：配置在计算机硬件平台上的第一层软件，是一组系统软件。（用户和计算机的接口）操作系统的工作就是管理计算机的硬件资源和软件资源，并组织用户尽可能方便地使用这些资源。主要特征：并发性、共享性、虚拟性、不确定性。操作系统的两个最基本的特征是程序并发性和资源共享性。操作系统**分为**批处理操作系统、实时操作系统和分时操作系统。嵌入式、个人计算机、网络分布式OS。
- **程序并发性的特征**：间断性、失去封闭性、不可再现性。
- （运行）-.阻塞（就绪）.进程调度.（运行）-I/O请求.（阻塞-I/O完成.（就绪）
- **进程的特性**：动态性、并发性、结构特性、独立性、异步性。
- **创建进程步骤**：申请分配一块PCB空间、为新进程分配资源、将数据从外存调入内存、把有关信息分别填入PCB的相应栏目中、把PCB插入就绪队列中。
- **死锁的条件**：互斥条件、不可抢占条件、占有且等待条件、循环等待条件。
- **虚存**：1）多次性.离散装配。一个作业将分成多次装入内存，部分装入便可启动执行；2）对换性。在进程运行期间，允许将那些暂不使用的程序和数据内存调至外存的对换区，待需要时再将其调入内存；3）虚拟性。虚拟存储器只是在逻辑上扩充内存容量，而实际上并没有真正扩大。
- **管程的特点**：外部过程不能访问局部数据变量；一个进程通过调用管程的一个过程进入管程；只能有一个进程在管程中执行.任何其他进程都被阻塞。
- **挂起状态进程的 4 个特点**：进程不能立即执行。进程可能是或不是正在等待一个事件。如果是，阻塞条件不依赖于挂起条件，阻塞事件的发生不会使进程立即被执行。为了阻止进程执行，可以通过代理把这个进程置于挂起态，代理可以是进程自己，也可以是父进程或操作系统。
- **缓冲作用**：1.缓和CPU与I/O设备之间速度不匹配的矛盾.2.提高CPU、与I/O设备间的并行性.3.减少对CPU的中断次数.放宽CPU对中断响应时间的要求。
.固定分区方案简单但有缺点：限制了系统中活动进程的数量
小作业不能有效地利用分区空间
- **线程优点：**创建和终止的时间少于进程；在内线程间切换的时间少于进程；提高了不同执行程序间的通信效率
- **文件结构：**1）连续（顺序）：支持顺序存取和随机存取，顺序读写速度快，磁盘寻道次数和时间少；不易扩展，存储空间利用率低会产生难以利用的磁盘碎片。2）链接（隐式）：读入块逻辑块，需要（i+1）次磁盘I/O。仅支持顺序访问，指向下个盘块的指针耗费少量存储空间；方便拓展，没有碎片问题，外存利用率高；（显示）一个磁盘甚至一张文件分配表，开机时将FAT读入内存并常驻内存，用户给出逻辑块号，操作系统从文件目录表找到对应FCB，从FCB算出物理块号，查找FAT，直接找到物理块，逻辑地址到物理地址的转化不需要I/O读写。3）索引：一个文件对应一张索引表；根据逻辑块号，找到该文件目录项FCB，从目录项找到索引表存放文职，将索引表从外存读入内存，查找索引表找到逻辑块在外存存放位置；支持随机访问，文件拓展容易实现，但索引表带来了系统开销。
- **进程互斥原则**：空闲让进，忙则等待，有限等待，让权等待。
- **银行家算法**是一种死锁避免算法，可以判断系统是否存在安全序列，无法判断系统是否处于死锁状态。
- **执行系统调用的过程：**1）传递系统调用参数：用户程序通过寄存器或者其他方式传递系统调用所需的参数。这些参数可能包括要执行的系统调用号以及其他参数，以便操作系统能够确定要执行的具体操作。2）执行陷入(trap)指令：系统调用过程首先需要执行陷入指令（也称为中断指令），用于触发从用户态切换到内核态的过程该指令将导致处理器从用户态切换到操作系统的特权模式。3）执行相应的服务程序：操作系统根据传递的系统调用号以及其他参数，选择相应的服务程序来执行所请求的操作。服务程序可能涉及对硬件设备的访问、文件系统操作、内存管理等特权操作。4)返回用户态：一旦服务程序执行完毕，操作系统将执行一些必要的清理操作并将控制返回给用户程序。此时，处理器从内核态切换回用户态，用户程序可以继续执行。
- **簇**：磁盘文件存储管理的**最小单位**。一个文件通常存放在一个或多个簇里，但至少要单独占据一个“簇”。也就是说两个文件不能存放在同一个簇中。簇（CLUST）指一组扇区(一个磁道可以分割成若干个大小相等的圆弧，叫扇区)的意思。

4.3 在进程概念中体现出的两个独立且无关的特点是什么？

资源所有权.进程包括存放进程映像的虚拟地址空间；回顾第 3 章的内容可知，进程映像是程序、数据、栈和进程控制块中定义的属性集。进程总具有对资源的控制权或所有权，这些资源包括内存、I/O通道、I/O 设备和文件等。操作系统提供预防进程间发生不必要资源冲突的保护功能。
调度/执行.进程执行时采用一个或多程序的执行路径（轨迹），不同进程的执行过程会交替进行。因此，进程具有执行态（运行、就绪等）和分配给其的优先级，是可被操作系统调度和分派的实体。

与读者/写者问题相关的条件通常有哪些？

存在一个多个进程共享的数据区，该数据区可以是一个文件，一块内存等；有些进程（reader）只能读取这个数据区中的数据，有些进程（writer）直往数据区中写数据。此外，还必须满足以下条件：任意数量的读进程可以同时读这个文件。一次只有一个写进程可以写文件。

对于多线程系统，这些元素中哪些可能属于线程控制块，哪些可能属于进程控制块？

进程标识：进程必须被标识，而进程中的每一个线程也必须有自己的 ID。
进程控制信息：调度和状态信息主要处于线程级；数据结构在两级都可出现；进程间通信和线程间通信都可以得到支持；特权在两级都可以存在；存储管理通常在进程级；资源信息通常也在进程级；
处理器状态信息：又名现场信息，一个进程在运行时存放在处理器现场中的各种信息。这些信息通常只与进程有关。
进程控制信息，用于管理和调度一个进程。常用的控制信息包括：
进程的调度相关信息，如进程状态、等待事件和等待原因、进程优先级、队列指引元等
进程组成信息，如正文段指针、数据段指针
引进程间通信相关信息，如消息队列指针、信号量等互斥和同步机制
进程在辅存储器内的地址、CPU 资源的占用和使用信息
如时间片余量、进程已占用 CPU 的时间、进程已执行的时间总和，记账信息。进程特权信息，如在内存访问和处理器状态方面的特权。

为何在使用虚拟内存时，局部性原理至关重要？

局部性原理描述了一个进程中程序和数据引用的集簇倾向。因此，假设在很短的时间内仅需要进程的一部分块是合理的。同时，还可以对将来可能会访问的块进行猜测，从而避免系统抖动。局部性原理表明虚拟内存方案是可行的。

5.4 列出进程间的三种互相知道的程度，并简要给出各自的定义。

进程之间相互不知道对方的存在：这是一些独立的进程，它们不会一起工作。关于这种情况的最好例子是多个独立进程的多道程序设计，可以是批处理作业，也可以是交互式会话，或者是两者的混合。尽管这些进程不会一 起工作，但操作系统需要知道它们对资源的竞争情况（competition）。例如，两个无关的应用程序可能都想访问同一个磁盘、文件或打印机。操作系统必须控制对它们的访问。
进程间接知道对方的存在：这些进程并不需要知道对方的进程 ID，但它们共享某些对象，如一个 I/O 缓冲区。这类进程在共享同一个对象时会表现出合作行为（cooperation）。
进程直接知道对方的存在：这些进程可通过进程 ID 互相通信，以合作完成某些活动。同样，这类进程表现出合作行为。
父进程与子进程：父进程与子进程可以并发执行；**不能**共享虚拟地址空间；父进程与子进程有不同的进程控制块；可以同时使用同一临界资源（通过各种同步机制实现对该资源的互斥访问。）
两个进程的共享段：在物理内存中仅保存一份段S的内容；P1和P2共享段S在共享段表中的段表项；P1和P2都不再使用段S时才回收段S所占的内存空间，S在P1和P2中的段号可以不同。

杂七杂八的

- 若干事件在同一时刻发生称为 **并行**；在同一时间间隔内发生称为**并发**。（相当于单/多处理器为发行）• 并行的事件一定是并发的，并发的事件或活动未必是并行的。并行性是并发行的特例，而并发是并行性的扩展。• 在同一时间内只能被一个进程访问的资源称为**临界资源**，每个进程中访问临界资源的那段程序称为**临界区**（互斥）。• 处理机在执行系统程序时的状态称为**系统态**，在执行用户程序时的状态称为**用户态**。• 分配资源以**进程**为单位。进程+执行上下文+程序+数据，操作系统通过PCB管理进程。• 虚拟内存的基础是**局部性理论**，虚存的最大容量一般由CPU的地址长度决定。虚存的实际容量由CPU的地址长和外存的容量决定，一般情况下，CPU的地址长度能表示的大小都大于外存容量。
- **原语**在执行过程中不能被中断，其代码应尽量短。阻塞原语(P)唤醒原语(V)
- 页式管理中，页的**划分**是由系统完成的；段式管理中，段的划分是由程序员完成的。
- 进程的逻辑地址到物理地址的转换，称为**重定位**：把用户程序装入内存时，对有关指令的逻辑地址部分的修改称为**地址重定位**。• 分页管理主存的分配是以页框为单位进行，逻辑地址分配是按页为单位进行分配，分配的内存页可以不连续。• 内存以**字节**为最小寻址单位，每个字节都有唯一地址。• 段式管理中，如果两个以上的进程都访问同一个段，则此段称为**共享段**。• 段页式管理得到物理地址要经过**三次访问**：第一次访问段表，得到页表首地址；第二次访问页表得到页面号；第三次将页面号和页内地址组合，得到物理地址。• 用户调用建立和打开文件操作来申请对文件的使用权。• 主存储器(内存)与外围设备之间的信息传送操作称为**输入输出操作**。• 多道程序设计的**硬件基础**是中断和通道。
- **线程间的模式切换**比进程间的模式切换开销更低的原因：包含的状态信息更少
- 单用户多处理系统中使用**线程**场景：前台和后台操作 、异步处理 、 加速执行 、模块化程序结构
- I/O指令只能在内核态执行，开关中断都是特权指令。
- **二级页表的分页系统：**页表被划分为两级结构：一级页表和二级页表。一级页表包含多个页表项，每个页表项对应一个二级页表的起始物理地址。每个二级页表包含多个页表项，用于映射虚拟页到物理页。CPU页表基址寄存器 (Page Table Base Register, PTBR) 存储的是当前进程的一级页表的起始物理地址通过在物理地址空间中访问该寄存器中的内容，CPU可以获取当前进程的一级页表，进而进行页表查找和地址映射操作。
- 磁盘扇区的**划分**是在磁盘的物理格式化操作中完成的。文件系统根目录的建立是在逻辑格式化操作中完成的
- 被中断程序的中断点由中断隐指令完成（不是系统调用或者操作系统）
- 用户级线程可以在不支持内核级线程的操作系统上实现
- 操作系统辅助将逻辑地址变为（逻辑块号，块内偏移量）的形式，并负责实现逻辑块号到物理块号的映射。

算法	思想	适用调度	是否抢占	优缺点	是否饥饿
先来先服务(FCFS)	按照进程到达先后	作业/进程	不可	简单，作业平均带权周转时间长，对短作业不友好	不会
短作业/进程优先（SJF/SPF）	要求最短进程先	作业/进程	通常不可，但是最短剩余时间优先算法(SRTN)抢占	平均周转时间通常最短	会，长作业可能会饿死
高相应比优先(HRRN)	每次调度前先计算相应比，高的先	作业/进程	不可抢占	综合考虑了等待时间和服务时间，不会饥饿	不会
轮转调度(RR)	每个进程最多执行一个时间片长度，没执行完就保存结果，并窜访到就绪队列，接着执行下一个	进程	抢占式	非常公平，有利于人机交互	不会
优先级调度	按照优先级(静态or动态)	进程	根据需求	考虑了进程的紧迫性	会，优先级大低可能会
多级反馈队列	设置多个就绪队列，优先级一次递减，时间片一次递增。每个队列采用FCFS，若该时间片未完成，则加入下一队列，若排到最后一个队列，则采用RR算法。若执行低优先级任务时来了高优先级任务，则暂停并执行高优先级任务。	进程	抢占式	不必事先知道进程的所需执行时间，但是复杂	通常不会

算法	算法思想	分区排列顺序	优点	缺点
首次适应	从头到尾寻找合适的分区	空闲分区以地址递增的次序排列	综合最好，开销小，回收分区后一般不需对空闲分区队列重新排列	
最佳适应	优先使用更小的分区，以保留更多的大分区	空闲分区以容量递增的次序排列	会有更多的大分区保留，更能满足大进程需求	产生太多小、难以利用的碎片；开销大，回收分区后可能需对空闲分区队列重新排列
最坏适应	优先使用更大的分区，以防产生太小的不可用碎片	空闲分区以容量递减的次序排列	可以减少难以利用的小碎片	不利于大进程；开销大
邻近适应	每次从上次查找结束的位置开始查找	空闲分区以地址递增的次序排列	不用每次都从低地址开始检索，算法开销小	会使高地址的大分区也被用完

	硬链接	软链接
是否持有目标文件	是	否，仅为“快捷方式”
目标文件类型	必须为文件实体	文件或文件夹
是否共享索引结点	是	否，有独立的索引结点
引用计数	创建文件加1，反之减一	恒为1
目标文件访问速度	快	需要根据目标文件路径访问，相对较慢

先来先服务(FCFS)：根据进程请求访问磁盘的先后次序进行调度。优点：公平，性能还可以；缺点：如果有大量进程竞争使用磁盘，请求访问磁道很分散，则 FCFS 在性能上很差，寻道时间长。**最短寻道时间优先（SSTF）：**其要求访问的磁道与当前磁头所在的磁道距离最近，以使每次的寻道时间最短，但这种调度算法却不能保证平均寻道时间最短。优点：性能较好，平均寻道时间短；缺点：可能产生“饥饿”现象。**扫描算法（SCAN）：**当磁头正在由里向外移动时，SCAN 算法所选择的下一个访问对象应是其欲访问的磁道，既在当前磁道之外，又是距离最近的。这样由里向外地访问，直至再无更外的磁道需要访问时，才将磁臂换向，由外向里移动。也叫电梯算法。优点：性能较好，平均寻道时间短，不会产生饥饿现象；缺点：1，只有到最边上的磁道才能改变磁头的移动方向。2.SCAN 对于各个位置磁道响应频率不平均。**循环扫描算法（CSCAN）：**为了减少 SCAN 算法造成的某些进程的请求被严重推迟，CSCAN 算法规定磁头单向移动：优点：比起 SCAN 算法，对于各位置磁道的响应频率很平均。缺点：只有到边上才能改变磁头移动方向，比起 SCAN 算法来，平均寻道时间更长

页式存储管理允许用户的编程空间为32个页面（每页1KB），主存为16KB。如有一用户程序为10页长，且某时刻该用户程序页表见右表。

若分别遇到3个逻辑地址0AC5H，1AC5H，3AC5H处的操作，计算说明系统将如何处理

逻辑页号	物理块号
0	8
1	7
2	4
3	10

Solution:

页面大小为1KB，所以低10位为页内偏移地址；用户编程空间为32个页面，即逻辑地址高5位为页号；主存为16个页面，即物理地址高4位为物理块号

0AC5H转换为二进制：000 1010 1100 0101B，页号为2，映射物理块号4，系统访问物理地址：12C5H（01 0010 1100 0101B）

1AC5H转换为二进制：001 1010 1100 0101B，页号为6，不在页表中，进行缺页中断处理

3AC5H转换为二进制：011 1010 1100 0101B，页号为14，用户程序只有10页，产生越界中断

例题

1.某文件系统中，针对每个文件，用户类别分为4类：安全管理员、文件主、文件主的伙伴、其他用户；访问权限分为5种：完全控制、执行、修改、读取写入。若文件控制块中用二进制位串表示文件权限，为表示不同类别用户对一个文件的访问权限，则描述文件权限的位数至少应为$4*5=20$位（不是5位因为这些权限是独立但不是互斥，要同时控制的，四类用户也要同时控制）

2.假设磁盘块大小位1KB，一个索引表项占4B，则一个磁盘块只能存放256个索引项，文件采用n层索引，文件最大长度为（256）^{n}*1KB。采用K层索引结构，且顶级索引表未调入内存，则访问一个数据块只需要$n+1$次读磁盘操作。

3. 某计算机主存按字节编址，逻辑地址和物理地址都是32位，页表项大小为4字节。请回答下列问题。

1）若使用一级页表的分页存储管理方式，逻辑地址结构为：页号（20位）+页内偏移量（12位），则页的大小是多少字节？页表最大占用多少字节？

因为该计算机主存按字节编址，页内偏移量是12位，所以页大小是2^{12}×1B=4KB

页号占20位，说明最多有2^{20}个页面，一个页表项占4B，一级页表最大占用

2）采用(1)中的分页存储管理方式，一个代码段起始逻辑地址为0000 8000H，其长度为8KB，被装载到从物理地址0090 0000H开始的连续主存空间中。页表从主存0020 0000H开始的物理地址处连续存放，如下图所示（地址大小自下向上递增）。请计算出该代码段对应的两个页表项的物理地址、这两个页表项中的页框号以及代码页面2的起始物理地址。

一个页面占4kb，需要两个连续页面存放。页表起始地址为00200000H，页表项物理地址=页表起始地址+页号*页表项字节数，第一个页面页号为8，第二个页面页号为9。8*4B=32B=0020H，页面1和2的物理地址分别为00200020H和00200024H。代码的起始物理地址为00900000H，前5位为页框号，则因为页面1为00900000H，所以页面2为00901000H（多一个页框号）

4. 某文件系统采用索引节点存放文件的属性和地址信息，簇大小为4KB。每个文件索引节点占64B，有11个地址项，其中直接地址项8个，一级、二级和三级间接地址项各1个，每个地址项长度为4B。请回答下列问题。

（1）最大文件长度：4KB/4B=1024，最大为（8+1024+1024*1024+1024*1024*1024）*4KB

一个文件系统中，FCB 占64B，一个盘块大小为1KB，采用一级目录，假定文件目录中有3200个目录项，则查找一个文件平均需要_____次访问磁盘

一个文件目录项对应一个文件控制块，我们查找一个文件是不是查找它的目录项即可，顺序查找目录表平均需要查找1600次(n个元素的顺序表平均查找次数为(n+1)/2)，一个磁盘块大小为1KB，一个文件控制块大小为64B，一个磁盘块中有1KB/64B=16个文件控制块，相当于查找了1600/16=100个磁盘

某银行提供 1 个服务窗口和 10 个顾客等待座位。顾客到达银行时，若有空座位，则到取号机领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下：

```
semaphore mutex = 1;   // 座位区互斥信号量
semaphore empty = 10;  // 座位区空闲位置数量
semaphore full = 0;    // 座位区等待人的数量
semaphore service_i = 0; // 顾客i等待叫号
semaphore machine = 1; // 取号机

cobegin
process 顾客i {
    P(empty);
    P(machine);
    从取号机获得一个号码;
    V(machine);
    P(mutex);
    一位顾客进入座位区;
    V(mutex);
    V(full);
    P(service_i);   // 等待叫号
    获得服务;
}
process 营业员 {
    while (TRUE) {
        P(full);
        P(mutex);
        一位顾客离开座位区;
        V(mutex);
        V(empty);
        V(service_i);
        叫号;
        为顾客服务;
    }
}
Coend
```

假设计算机系统采用CSCAN（循环扫描）磁盘调度策略，使用2KB的内存空间记录16384个磁盘块的空闲状态。

(1) 请说明在上述条件如何进行磁盘块空闲状态的管理。

(2) 设某单面磁盘的旋转速度为6000rpm，每个磁道有100个扇区，相邻磁道间的平均移动的时间为1ms。若在某时刻，磁头位于100号磁道处，并沿着磁道号增大的方向移动（见下图），磁道号的请求队列为50, 90, 30, 120，对请求队列中的每个磁道需读取1个随机分布的扇区，则读完这个扇区点共需要多少时间？需要给出计算过程

某32位系统采用基于二级页表的请求分页存储管理方式，按字节编址，页目录项和页表项长度均为4字节，虚拟地址结构如下所示。

页目录号（10位）	页号（10位）	页内偏移（12位）

某C程序中数组a[1024][1024]的起始虚拟地址为1080 0000H，数组元素占4字节，该程序运行时，其进程的页目录起始物理地址为0020 1000H，请回答下列问题。

(1) 数组元素a[1][2]的虚拟地址是什么？对应的页目录号和页号分别是什么？对应的页目录项的物理地址是什么？若该目录项中存放的页框号为00301H，则a[1][2]所在页对应的页表项的物理地址是什么？

(2) 数组a在虚拟地址空间中所占区域是否必须连续？在物理地址空间中所占区域是否必须连续？

(3) 已知数组a按行优先方式存放，若对数组a分别按行遍历和按列遍历，则哪一种遍历方式的局部性更好？

解答：

(1) 第一问。数组元素a[i][j]的虚拟地址 = 数组a的起始虚拟地址+(i×数组a的行数+j)×数组元素占地址单元数。数组a[1024][1024]的起始虚拟地址为1080 0000H，行数为1024，每个数组元素占4字节，按字节编址，每个数组元素占4个地址单元。数组元素a[1][2]的虚拟地址 = 1080 0000H + (1024×1+2)×4=1080 1008H。

第二问。数组元素a[1][2]的虚拟地址1080 1008H =0001000001000000000010000000001000

页目录号 页号 页内偏移量

B，对应的页目录号是0001000010B=042H，页号是00 0000 0001B=001H。

第三问。因为该程序运行时，其进程的页目录起始物理地址为0020 1000H，页目录项长度为4字节，按字节编址，每个页目录项占4个地址单元，数组元素a[1][2]所在页目录号为042H，所以对应的页目录项的物理地址是0020 1000H+4×042H=0020 1108H。

第四问。因为该目录项中存放的页框号为00301H，页表项长度为4字节，按字节编址，每个页表项占4个地址单元，数组元素a[1][2]对应页号是001H，页内偏移为4×001H=004H，根据虚拟地址结构，页内偏移占12位，拼接页框号00301H和页内偏移004H得到a[1][2]所在页对应的页表项的物理地址为00301004H。

(2) 第一问。因为数组要求支持随机访问，所以数组a在虚拟地址空间中所占区域必须连续。

第二问。因为系统采用请求分页存储管理方式，操作系统使用地址映射技术将虚拟地址映射到物理地址，这种映射关系可以是灵活的，不一定要求数组a在物理内存中占据连续的区域。所以数组 a 在物理地址空间中所占区域不必连续。

(3) 因为数组a在虚拟地址空间连续存放，系统采用请求分页存储管理方式，操作系统使用地址映射技术将虚拟地址映射到物理地址，数组a在同一页框中的元素连续存放，所以当访问方式和数组存放方式一致时，对数组遍历局部性更好。因为数组a按行优先方式存放，所以对数组a按行遍历局部性好。

某一页式系统，其页表放在主存中：

a) 若对主存的一次存取需要1.5 us，问实现一次页面访问时存取时间是多少？

b) 若系统有快表且平均命中率为85%，而页表在快表中的查找时间可以忽略不计，试问此时的存取时间为多少？

Solution:

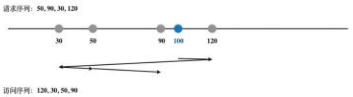
页表在主存，实现一次存取需要访问两次主存：第一次是访问页表获取所需访问数据所在页面的物理地址；第二次是根据物理地址存取数据

a) CPU 必须访问主存两次：1.5 × 2 = 3us

b) 增加快表后，在快表中找到页面的概率为85%，存取时间为：0.85 × 1.5 + 0.15 × (1.5 + 1.5) = 1.725us

(1) 用位图表示磁盘块的空闲状态，每位表示一个磁盘块的空闲状态。若该位为0，则该位对应的磁盘块空闲；若该位为1，则该位对应的磁盘块不空闲。总共有16384个磁盘块，需要16384bit = 2^{14} bit = 2KB的内存空间。

(2) 初始时，磁头位于100号磁道处，沿着磁道号增大的方向移动，采用CSCAN（循环扫描）磁盘调度策略，这里没有给出最小磁道号和最大磁道号，CSCAN视为CLOOK，磁道号的请求队列为50, 90, 30, 120，模拟过程如下：



访问序列为：120, 30, 50, 90，总体分三段，第一段100号磁道处移动到120号磁道处，移动磁道数为120-100=20，第二段120号磁道处移动到30号磁道处，移动磁道数为120-30=90，第三段30号磁道处移动到90号磁道处，移动磁道数为90-30=60，总的移动磁道数为20+90+60=170。相邻磁道间的平均移动的时间为1ms。所以磁道移动时间为170×1ms=170ms。

磁盘转速为6000rpm=100r/s，每访问一个扇区的平均旋转延迟时间为1/(2×100r/s)=5ms，需要访问4个扇区，总计旋转延迟时间为5ms×4=20ms。

每个磁道有100个扇区，所以一个扇区占0.01r，读取一个扇区时间为0.01r/(100r/s)=0.1ms。需要访问4个扇区，总计磁盘读取时间为0.1ms×4=0.4ms。

访问这些扇区时间=磁道移动时间+旋转延迟时间+磁盘读取时间

=170ms+20ms+0.4ms=190.4ms。

##

例题

7.2 为何需要**重定位进程**的能力？

在多道程序设计系统中，可用的内存空间通常被多个进程共享。通常情况下，程序员事先并不知道在某个程序执行期间会有其他哪些程序驻留在内存中。此外，我们还希望提供一个巨大的就绪进程池，以便把活动进程换入或换出内存，进而使处理器的利用率最大化。程序换出到磁盘中后，下次换入时要放到与换出前相同的内存区域会很困难。相反，我们需要把进程重定位（relocate） 到内存的不同区域。

7.3 为何**不可能在编译时实施内存保护**？

由于程序在内存中的位置不可预测，因而在编译时不可能检查绝对地址来确保保护。此外，大多数程序设计语言允许在运行时进行地址的动态计算。因此必须在运行时检查进程产生的所有内存访问。

7.4 允许**两个或多个进程访问内存某一特定区域**的原因是什么？

任何保护机制都必须具有一定的灵活性，以允许多个进程访问内存的同一部分。例如，多个进程正在执行同一个程序时，允许每个进程访问该程序的同一个副本，要比让每个进程有自己单独的副本更有优势。合作完成同一个任务的进程可能需要共享访问相同的数据结构。因此，内存管理系统在不损害基本保护的前提下，必允许对内存共享区域进行受控访问。我们将会看到用于支持重定位的机制也支持共享。

8.5 **转换检测缓冲区**的目的是什么？

原则上，每次虚存访问都可能会引起两次物理内存访问：一次取相应的页表项，另一次取需要的数据。因此，简单的虚拟内存方案会导致内存访问时间加倍。为克服这个问题，大多数虚拟内存方案都为页表项使用了一个特殊的高速缓存，通常称为转换检测缓冲区（TranslationLookaside Buffer, TLB）。

8.6 简单定义**两种可供选择的页面读取策略**。

请求分页，只有当访问到某页中的一个单元时才将该页取入内存。若内存管理的其他策略比较合适，将发生下述情况：当一个进程首次启动时，会在一段时间出现大量的缺页中断：取入越来越多的页后，局部性原理表明大多数将来访问的页都是最近读取的页。因此，在一段时间后错误会逐渐减少，缺页中断的数量会降到很低。

预先分页，读取的页并不是缺页中断请求的页。预先分页利用了大多数辅存设备（如磁盘）的特性，这些设备有寻道时间和合理的延迟。若一 个进程的页连续存储在辅存中，则一次读取许多连续的页要比隔一段时间读取一 页有效。 当然，若大多数额外读取的页未引用到，则这个策略是低效的。

8.9 **页缓冲**实现什么功能？

为提升性能，不丢弃置换出的页，被置换出的页任然驻留在内存中。因此，若进程访问该页，则可迅速返回该进程的驻留集，且代价很小。

8.12 请求式清除和预约式清除有何区别？

请求式清除（demand cleaning），只有当一页被选择用于置换时才被写回辅存；

预约式清除（precleaning）策略则将这些已修改的多页在需要使用它们所占据的页框之前成批写回辅存。

×4B=4MB。