

操作系统背诵笔记

操作系统概述

QQ为 1198285594 的学长免费分享

- 为什么多道批处理系统可以提高资源利用率？（2013 922）p8
 1. 提高 CPU 利用率：在内存中装有若干道程序，使其交替运行，CPU 始终处于忙碌状态；
 2. 提高内存和 I/O 设备利用率：允许并发执行，使内存和 I/O 处于忙碌状态；
 3. 增加系统吞吐量：内存和 I/O 设备不断忙碌，必然会增加系统的吞吐量。
- 在多道批处理系统中，是否系统中并发的进程越多。资源利用率越好，为什么？（2016 922 829）

不是，并发的进程越多，调度的时空开销越大，所以并不是进程越多，并发性越好。

QQ为 1198285594 的学长免费分享

QQ为 1198285594 的学长免费分享

进程管理

- 实现进程同步机制必须遵循哪几条准则，含义是什么？（2012 922）（2014 922）p50
 1. 空闲让进：临界区空闲，应允许一个请求进入临界区的进程立即进入临界区。
 2. 忙则等待：已有进程进入临界区，其它请求进入临界区的进程必须等待。
 3. 有限等待：对要求访问临界区的进程，应使其在有限的时间内能进入自己的临界区。
 4. 让权等待：当进程不能进入自己的临界区，因立即释放处理机，以免进程进入“忙等”状态。
- 为什么要引入线程，线程和进程有何区别？（2013 922 829）p72、73
 1. 减少程序在并发执行时所付出的时空开销。
 2. 使 OS 具有更好的并发性。
- 1. 调度：同一进程中，线程的切换不会引起进程的切换，一个进程中的线程切换到另一个进程中的线程，会引起进程切换。

2. 并发性：线程和进程都可以并发执行。
3. 拥有资源：线程自己不拥有资源（也有一点必不可少的资源），但它可以访问其隶属进程的资源。
4. 系统开销：线程切换的开销小于进程切换的开销。

➤ 从操作系统设计的角度谈谈进程控制块的作用？（2014 922 829）

使一个在多道程序环境下不能独立运行的程序，成为一个能独立运行的基本单位，一个 能与其它进程并发执行的进程，PCB 是进程存在的唯一标志。

➤ 简述消息缓冲队列通信机制。（2014 922 829）

发送原语利用 send 语句将消息直接发送给接收进程；接收进程利用 receive 语句接收消息

➤ 系统型线程和用户型线程有何区别？（2017 922 829）

1. 开销上：系统型进程由于在切换时要从用户态转换到核心态，模式切换导致开销较大，用户级线程在切换时不需要转换到内核空间，开销小。
2. 进行系统调用时：用户型线程会被阻塞，而且进程内的所有线程都会被阻塞。而在系统型线程中，进程中的其他线程任然可以运行。
3. 在多核处理机上：采用用户型线程的应用不能利用多处理机进行多重处理。而系统型线程可以充分利用多核，每一个核上并行运行线程，提高了系统的执行速度和效率。

➤ 画出引入挂起和激活机制后，进程状态转换图。（2018 922）

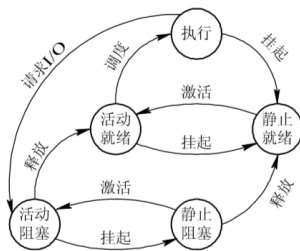


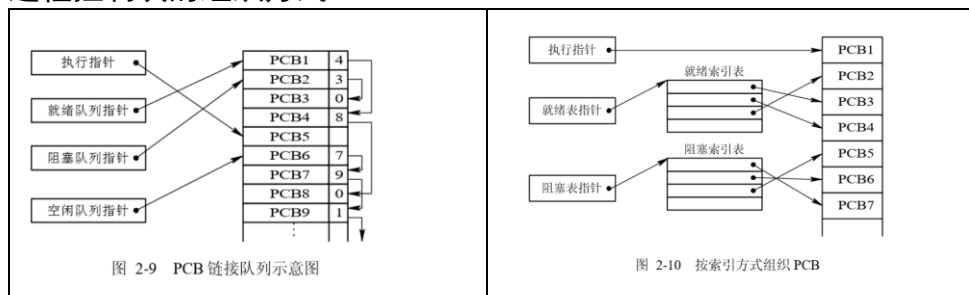
图 2-6 具有挂起状态的进程状态图

预测：

● 进程

1. 进程是一个可并发实行的具有独立功能的程序关于某个数据集合的一次执行过程，也是操作系统进行资源分配和保护的基本单位。
2. 线程是轻量级进程，是进程中的一条执行路径。
3. 引入进程的目的：使多个进程能够并发执行，以提高资源利用率和系统吞吐量。
4. 程序并发执行的特征：间断性；失去封闭性；不可再现性
5. 进程的实质是进程实体的一次执行，“动态性”是进程最基本的特征。

6. 创建状态：进程已经拥有了自己的 PCB，但进程自身还未进入主存，进程还不能被调度执行，其所处的状态就是创建状态。
7. 进程控制块的组织方式：



8. 引起创建进程的事件：用户登录、作业调度、提供服务、应用请求
9. 进程的创建：申请空白 PCB、为新进程分配资源、初始化进程控制块、将新进程插入就绪队列。
10. 进程同步的主要任务是：对多个先关进程在执行次序上进行协调，以使并发执行的诸进程之间能有效地共享资源和相互合作，从而使程序的执行具有可再现性。
11. AND 同步机制的基本思想：将进程在整个运行过程中需要的所有资源，一次性全部地分配给进程，待进程使用完后再一起释放。只要尚有一个资源未能分配给进程，其它所有能为之分配的资源也不分配给它。
12. 管程：代表共享资源的数据结构，以及由对该数据结构实施操作的一组过程所组成的资源管理程序，共同构成了一个操作系统的资源管理模块，我们称之为管程。
13. 管程由四部分组成：管程名称、局部于管程内部的共享数据结构说明、对该数据结构进行操作的一组过程、对局部于管程内部的共享数据设置初始值的语句。
14. 高级通信类型：共享存储系统、消息传递系统、管道通信系统

调度与死锁

QQ为 1198285594 的学长免费分享

➤ 处理机调度有那三个层次？（2012 922）（2018 922 各自的主要任务是什么？）

1. 高级调度（作业调度）
目的：根据某种算法，决定将外存上处于后备队列中的哪些作业调入内存，为他们创建进程分配必要的资源并将其放入就绪队列；
2. 中级调度（中程调度）
目的：提高内存利用率和系统吞吐量
把暂时不能运行的进程调至外存等待；
3. 低级调度（进程调度）
目的：根据某种算法，决定将就绪队列中的哪个进程应获得处理机，并由分派程序将处理机分配给被选中的进程。

➤ 产生死锁的主要原因是什么？（2012 922）

1. 竞争资源
2. 进程推进顺序非法

➤ 有哪些处理死锁的基本方法？静态分配资源的方法属于哪种处理死锁的方法？而银行家算法属于哪种处理死锁的方法？（2012 922）

1. 预防死锁：破坏死锁产生的四个必要条件中的一个或者几个
2. 避免死锁：在资源动态分配时，用某种方法来防止系统进入不安全状态（银行家算法）
3. 检测死锁：在系统发生死锁时检测死锁的发生
4. 解除死锁：与检测死锁相配套，用于解除死锁。

➤ 什么是死锁定理？（2015 922）

S 为死锁的充分必要条件是：当且仅当 S 状态的资源分配图是不能完全简化的。该充分条件被称为死锁定理。

➤ 多级反馈队列调度算法是如何工作的？（2017 922）

1. 设置多个就绪队列，为每个队列赋予不同的优先级，优先级逐个队列递减。
2. 赋予各个队列中进程执行时间片不同的大小，队列优先级越高，时间片越小
3. 对于新进程，先将其放入第一级优先级队列的末尾，按 FCFS 算法调度，若在第一个时间片执行完，便撤离；若未完成，则将其放在第二级队列末尾按 FCFS 调度，若该进程进入最后一级队列，则在最后一级队列中按照时间片轮转方式执行，直至完成。
4. 当且仅当第一级队列为空时，调度程序才调度第二级队列中的进程运行，以此类推；如果 CPU 在执行某进程时，更高优先级队列中有了新进程，则 CPU 立即被抢占。

➤ 周转时间和带权周转时间的区别，为何引入带权周转时间？（2017 922）

周转时间=作业结束时间-提交时间；

带权周转时间=（作业结束时间-作业提交时间）/作业实际运行时间；

引入带权周转时间，使得不同长度作业的调度可以比较，带权周转时间越接近 1，说明进程的执行效率越高。

预测：

1. 选择调度方式面向系统的准则：系统吞吐率高、处理机利用率好、各类资源平衡利用。
2. FCFS：
有利于长作业，不利于短作业；有利于 CPU 繁忙型进程，不利于 I/O 繁忙型进程
3. 短作业（进程）优先调度算法 SJ(P)F
优点：有效降低系统平均等待时间，提高系统吞吐量；
缺点：对长作业不利、完全未考虑作业紧迫程度、作业长短由用户决

定不一定真正能做到短作业优先调度

4. 死锁：是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵局时，若无外力作用，它们都将无法前进。
5. 产生死锁的必要条件：
 - 1) 互斥条件：一段时间内某资源只由一个进程占用。
 - 2) 请求和保持条件：指进程已经保持了至少一个资源，但又提出了新的资源请求，而该资源又被其他进程占有，此时请求进程阻塞，但又对自己已经获得的资源保持不放。
 - 3) 不剥夺条件：指进程已经获得的资源，在未使用完之前，不能被剥夺，只能在使用完之后自己释放。
 - 4) 环路等待条件：发生死锁时，必然存在一个进程---死锁的环形链。
6. 安全状态：系统能够按照某种顺序，为每个进程分配所需的资源，直至满足每个进程对资源的最大需求。
7. 死锁的解除：
 - 1) 剥夺资源
 - 2) 撤销进程

内存管理

- 分页分段属于离散型的存管理方式，相对于连续内存管理方法的主要优点是什么?(2012 922)

1. 提高内存利用率；
2. 便于内存的动态申请和释放；
3. 便于信息共享，适用于多道程序；

- 给定段号和段内地址，完成段式管理中的地址变换过程（并用图表示）(2012 922)（2015 922）

在进行地址变换时，系统将逻辑地址中的段号与段表长度 TL 进行比较。若 $S > TL$ ，表示段号太大，是访问越界，于是产生越界中断信号；若未越界，则根据段表的始址和该段的段号，计算出该段对应段表项的位置，从中读出该段在内存的起始地址，然后，再检查段内地址 d 是否超过该段的段长 SL 。若超过，即 $d > SL$ ，同样发出越界中断信号；若未越界，则将该段的基址 d 与段内地址相加，即可得到要访问的内存物理地址。

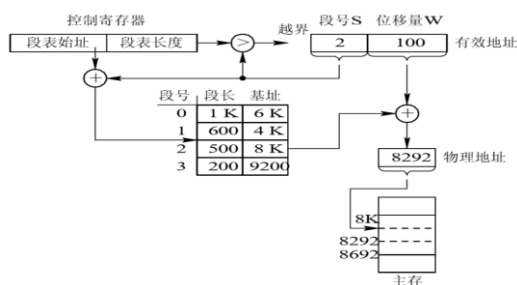


图 4-18 分段系统的地址变换过程

- 简述分页式虚拟存储系统中，一个逻辑地址到物理地址的转换过程（并画出

转换机构图)。(2013 922 829)(2014 922 829)

当进程要访问某个逻辑地址中的数据时,分页地址变换机构会自动地将有效地址(相对地址)分为页号和页内地址两部分,再以页号为索引去检索页表。在执行检索之前,先将页号与页表长度进行比较,如果页号大于或等于页表长度,则表示本次所访问的地址已超越进程的地址空间。于是产生地址越界中断。若未出现越界,则将页表始址与页号和页表项长度的乘积相加,便得到该表项在页表中的位置,于是可从中得到该页的物理块号,将之装入物理地址寄存器中。与此同时,再将有效地址寄存器中的页内地址送入物理地址寄存器的块内地址字段中。

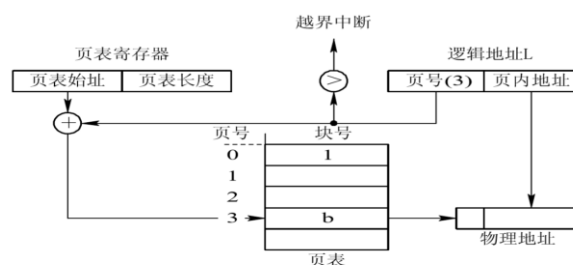


图 4-13 分页系统的地址变换机构

- 说明 LRU 相比 FIFO 算法有何优点。(2013 922)

FIFO 算法的性能差是因为它根据各个页面调入内存的时间,而页面导入的先后并不能反映页面的使用情况,而 LRU 是根据页面调入内存的使用情况来决策的,可以提高命中率,减少缺页中断。

- 解释静态链接和动态链接是现代操作系统中两种重要的链接方式,试比较同一程序经过静态链接和动态链接后的可执行个文件的大小,如果有不同分析原因。(2014 922 829)

静态链接:在程序运行之前,先将各目标模块及它们所需的库函数链接成一个完整的装配模块,以后不再拆开。

动态链接:

- 装入时动态链接:将用户源程序编译得到的一组目标模块,在装入时采取边装入边链接的方法。
- 运行时动态链接:对于某些目标模块的链接,是在程序执行中需要该模块时才进行链接。

静态链接产生的可执行文件大于动态链接,因为静态链接包含了运行所需的全部代码,而动态链接只是在文件中加入了调用的函数所在的文件模块和调用函数在文件中的位置等信息。

- LRU 页面置换算法的基本思想是什么?(2014 922 829)(2015 922 829 思想和依据)

按照页面在内存中的使用情况进行决策,选用最近最久未使用的页面予以淘汰。

依据:调入内存的页面的使用情况。

- 操作系统中虚拟存储器的基本原理是什么?(2012 922)

局部性原理

- 存取主存中的一条指令或数据至少要访问几次内存？如何提高速度？（2015 922 829）

2 次，使用块表 TLB 和高速缓存 Cache。

- 开发程序时用动态链接库有什么优点？（2016 922）

1. 编写程序体积小，占用空间少；
2. 减少页面交换，更加节省内存；
3. 使开发过程独立，耦合度小，适用于大规模的软件开发。

- 说明页面置换算法在虚拟存储管理中的重要性。（2016 922 829）（2017 922）

好的页面置换算法可以降低页面置换率，减少资源浪费，不好的页面置换算法会使页面发生抖动。

- FIFO 适用于什么场合，又有何缺点。（2016 922 829）

适用于按线性访问的地址空间，增加页面数可能出现缺页率不降反升的情况。

- 说明段页式系统中动态地址变换过程。（2016 922 829）

- 分段式系统和分页式系统有何区别？（2017 922 829）

页是数据存储的物理单位，分页是为了满足 OS 的管理需要，分段是为了满足用户的需要。

区别：页的大小固定并且由系统决定，段的大小不固定；分页的作业地址空间是一维的，分段的作业地址空间是二维的。

- 写出分页系统的地址转换过程。（2017 922 829）

- FIFO 和 LRU 算法各适用于什么场合？（2017 922）

FIFO 算法适合按线性顺序访问地址空间的情况；LRU 算法适合于频繁访问某些地址空间的情况。

- 内存中连续分配有何特点，为什么要引入离散分配？（2018 922）

连续分配会产生难以利用的碎片，要有连续的存储空间，必须知道文件的大小；引入离散分配是为了充分利用内存空间。

- 装入是时动态链接和运行时动态链接有何区别？哪种更节约内存？（2018 922）

装入时动态链接：将用户源程序编译得到的一组目标模块，在装入时采取边装入边链接的方法。

运行时动态链接：对于某些目标模块的链接，是在程序执行中需要该模块时才进行链接。

运行时动态链接更加节约内存。

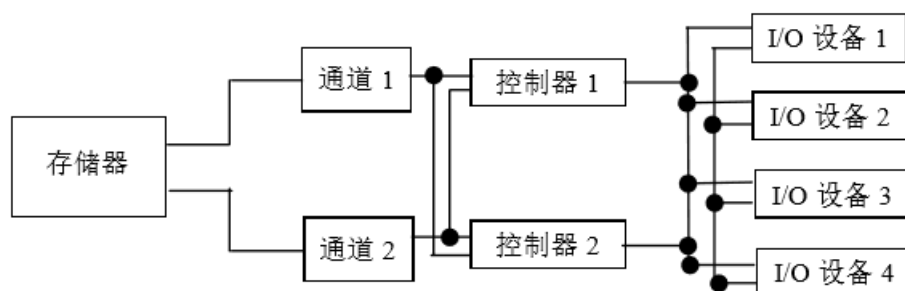
预测：

1. 虚拟存储器：具有请求调页和置换功能，能从逻辑上对内存容量进行扩充的一种存储系统。
2. 虚拟存储器的硬件支持：
 - a. 请求分页的页表（段表）机制
 - b. 缺页（段）中断机构
 - c. 地址变换机构
3. 虚拟存储器具有多次性、对换性、轮换性三大主要特征。
4. LRU 算法的硬件支持：寄存器或栈。

设备管理

➤ 什么是通道，通道经常采用如图所示的交叉连接，为什么？（2013 922 829）

(2016 922 什么是通道的瓶颈问题，如何处理此问题，画出示意图)



通道：通道是一种特殊的处理机，它具有执行 I/O 指令的能力，并通过执行通道（I/O）程序来控制 I/O 操作。

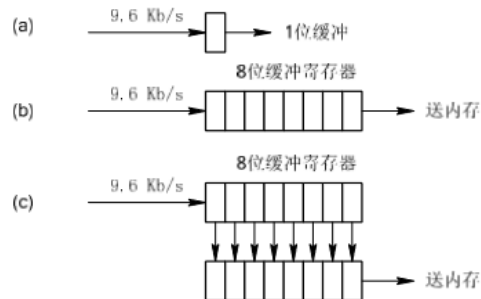
为了解决通道不足所引起的“瓶颈现象”，增加 I/O 设备到主机的通路，这样不仅解决了“瓶颈”现象，也提高了系统的可靠性。

瓶颈问题：由于通道价格昂贵，致使机器中所设置的通道数量减少，这往往有使它成了 I/O 的瓶颈，而造成整个系统吞吐量下降。

➤ 简述操作系统引入缓冲管理的原因？（2013 922 829）（2017 922 829 常见的缓冲模式）（2018 922）

1. 缓和 CPU 和 I/O 设备之间的速度不匹配，凡是数据到达率和离去率不同的地方，都可以设置缓冲。
 2. 减少对 CPU 的中断频率。
 3. 提高 CPU 和 I/O 设备之间的并行性。
- 常见缓冲模式：单缓冲和双缓冲、循环缓冲、缓冲池。

- 按照下图说明操作系统中引入缓冲的好处。(2015 922 829)



减少对 CPU 的中断频率，放宽对 CPU 中断响应时间的限制。在远程通信系统中，如果从远地终端发来的数据仅用一位缓冲来接收，如图所示，则必须在每收到一位数据时便中断一次 CPU，这样，对于速率为 9.6 Kb/s 的数据通信来说，就意味着其中断 CPU 的频率也为 9.6 Kb/s，即每 $100\ \mu\text{s}$ 就要中断 CPU 一次，而且 CPU 必须在 $100\ \mu\text{s}$ 内予以响应，否则缓冲区内的数据将被冲掉。倘若设置一个具有 8 位的缓冲(移位)寄存器，如图所示，则可使 CPU 被中断的频率降低为原来的 $1/8$ ；若再设置一个 8 位寄存器，如图所示，则又可把 CPU 对中断的响应时间放宽到 $800\ \mu\text{s}$ 。

- 缺页中断与其他普通中断的主要区别是什么？(2016 922)

共同点：都要保护 CPU 现场，分析中断原因，转换中断处理程序进行处理和恢复 CPU 等操作。

区别：在指令执行期间产生和处理信号；一条指令在执行期间可能产生多次缺页中断。

- 单缓冲下，为什么系统对一块数据的处理时间为 $\max(C, T) + M$ ？

在块设备输入时，假定从磁盘把一块数据输入到缓冲区的时间为 T ，操作系统将该缓冲区中的数据传送到用户区的时间为 M ，而 CPU 对这一块数据处理(计算)的时间为 C 。由于 T 和 C 是可以并行的(见图 5-11)，当 $T > C$ 时，系统对每一块数据的处理时间为 $M + T$ ，反之则为 $M + C$ ，故可把系统对每一块数据的处理时间表示为 $\max(C, T) + M$ 。

- 推动 I/O 发展的动力是什么，有哪几个发展阶段？(2016 922)

尽量减少主机对 I/O 的控制干预，把主机从繁杂的 I/O 控制事务中解脱出来，以便完成数据处理任务。

1. 程序中断方式：程序直接对特定设备进行测试，CPU 利用率低；
2. 中断驱动 I/O 控制方式：引入中断机制，当设备准备完成时发生中断，

提高 CPU 利用率。

3. DMA 方式:在 I/O 设备和主存之间开辟通路,极大提高了 CPU 利用率。
4. I/O 通道控制方式:引入专门的 I/O 处理机进行 I/O 管理,解放了 CPU。

➤ SPOOLING 技术如何实现,在操作系统中起何作用? (2017 922 829)

当有进程要输出时, SPOOLing 技术不是直接把机器的使用权交给处理机,而是做以下两件事情:

1. 由输出进程在输出井中为之申请一个空闲磁盘块区,并将要输出的数据放入其中。
2. 输出进程再为该进程申请一张空闲的请求输出表,并将进程的输出要求放入其中,再将该表送到请求输出队列中。如果还有进程要输出,则重复以上步骤。

特点:

1. 提高了 I/O 速度;
2. 将独占设备改造为共享设备;
3. 实现了虚拟设备功能;

预测:

1. 设备管理的基本任务:完成用户提出的 I/O 请求,提高 I/O 利用率以及提高 I/O 设备的利用率。
2. 设备管理的主要功能有:缓冲区管理、设备分配、设备处理、虚拟设备及实现设备独立性等。
3. I/O 通道与一般处理机不同之处在于:指令类型单一;没有自己的内存;
4. SPOOLing 系统主要组成:
 - 1) 输入井和输出井;
 - 2) 输入缓冲区和输出缓冲区 ;
 - 3) 输入进程和输出进程;
5. 磁盘调度的目标是使磁盘的平均寻道时间最少。
6. 提高磁盘 I/O 的其他方法:
 - 1) 提前读:对文件进行访问时,经常采取顺序访问,所以在读当前块的时候,把下一块也读入缓冲区。
 - 2) 延迟写:对于缓冲区中要写回磁盘的数据,并不立即写,而是将其挂在空闲缓冲区队列末尾,直至该块被申请时,才写入磁盘。
 - 3) 优化物理块分布。
 - 4) 虚拟盘:利用内存空间去仿真磁盘。
7. 虚拟盘与磁盘高速缓存的主要区别在于:虚拟盘中的内容完全由用户控制,而高速磁盘缓存中的内容则是由 OS 控制的。

磁盘与文件系统

➤ 磁盘访问时间由哪几部分组成? (2012 922) (2018 922 829)

寻道时间、旋转延迟时间、数据传输时间

- 常用的提高文件系统性能的方法有哪些？（2012 922）

提前读：对文件进行访问时，经常采取顺序访问，所以在读当前块的时候，把下一块也读入缓冲区。

延迟写：对于缓冲区中要写回磁盘的数据，并不立即写，而是将其挂在空闲缓冲区队列末尾，直至该块被申请时，才写入磁盘。

优化物理块分布。

虚拟盘：利用内存空间去仿真磁盘。

- 何谓文件的物理结构，可以分为哪几类，比较其优点？(2013 922 829)

文件的物理结构又称文件的存储结构，是指文件在外存上的存储组织方式。

1. 连续分配：

优点：顺序访问容易，顺序访问速度快；

缺点：要有连续的存储空间，必须事先知道文件的长度；

2. 链接分配：

优点：显著提高外存利用率，便于动态增长，增删改十分方便；

缺点：不能直接存取，FAT 需要大量的存储空间；

3. 索引分配：

优点：占用内存小，可以直接存取；

缺点：花费较多的外存空间

- 操作系统采用何种磁盘调度算法更好？（2013 922 829）

电梯调度算法，因为最短寻道时间算法容易造成饥饿现象。

- 试比较磁盘高速缓存和虚拟盘，提高文件系统性能通常有哪些方法？（2014 922）

磁盘高速缓存：利用内存中的存储空间来暂存从磁盘中读取的信息，是一组在逻辑上属于磁盘，在物理上驻留内存中的磁盘块。

虚拟盘：是指利用内存空间去仿真磁盘；

区别：虚拟盘中的内容完全由用户控制，而高速磁盘缓存中的内容则是由 OS 控制的。

- 举例说明线性检索过程（例如查找/usr/ast/mbox）(2014 922)（2015 922）

首先，系统应先读入第一个文件分量名 `usr`，用它与根目录文件(或当前目录文件)中各目录项中的文件名顺序地进行比较，从中找出匹配者，并得到匹配项的索引结点号，再从索引结点中得知 `usr` 目录文件存放的盘块，将该盘块内容读入内存。

接着，系统再将路径名中的第二个文件分量名 `ast` 读入，用它与放在刚刚读入内存盘块中的第二级目录文件中各目录项的文件名顺序进行比较，又找到匹配项，从中得到 `ast` 的目录文件存放的索引结点，再从该索引结点中得知 `/usr/ast` 存

放的盘块中，再将该盘块读入内存。

然后，系统又将该文件的第三个分量名 `mbox` 读入，用它与第三级目录文件 `/usr/ast` 中各目录项中的文件名进行比较，最后得到 `/usr/ast/mbox` 的索引结点，该索引结点中存放了指定文件的物理地址。目录查询操作到此结束。如果在顺序查找过程中发现有一个文件分量名未能找到，则应停止查找，并返回“文件未找到”信息。

- SSTF 磁盘调度算法有何缺点。（2014 922 829）

导致“饥饿”现象的发生。

- 给出描述文件的数据结构（即文件控制块）和文件目录结构；（2015 922 829）

文件控制块：用于描述和控制文件的数据结构；

文件与文件控制块一一对应，文件控制块的有序集合称为目录；

内容：

1. 基本信息类：文件名，文件物理位置，文件逻辑结构，文件物理结构；
2. 存取控制信息：用户的存取权限；
3. 使用信息类：文件的建立日期，使用日期。

文件目录结构：文件扩展名、文件长度、文件类型、文件物理地址、状态位；

- 以索引节点为文件系统的物理文件组织结构，图示索引节点结构，说明其优点；（2015 922 829）

文件名	索引节点编号
-----	--------

优点：将文件与文件描述信息分开，减少了查找信息时的 I/O 信息量，提高了检索效率。

- 为该文件系统设计几个必要的系统调用，选其中一个为例，详细说明实该系统调用的方法和过程（注意使用以上设计中的数据结构）。（2015 922）

创建文件、删除文件、读文件、写文件、打开文件、关闭文件；

例子：利用创建文件的系统调用 `create`，由系统根据用户提供的文件名和存取方式来创建一个新文件，当用户已不再需要某文件时，便可利用 `unlink` 将指定文件删除

- 相对于顺序文件和索引文件，索引顺序文件有何优点？如果在一个索引顺序文件中含有 n 个记录，如何设计索引顺序文件，令检索指定关键字记录的平均查找次数最少？（2018 922）

克服了变长记录文件不便于直接存取的特点，而且付出的代价不算太大，可以为顺序文件建立多级索引，索引表所占的系统开销更小。

- 如何应用 RAID（廉价磁盘冗余阵列）提高磁盘访问速度，请画图示意。（2018 922 829）

采用交叉存取技术，将每一盘块中的数据分为若干个子盘块数据，再把每一个子盘块的数据分别存储到各个不同磁盘中的相同位置上。当要将一个盘块的数

据传送到内存时，采取并行传输方式，将各个盘块中的子盘块。数据同时向内存中传输，从而使传输时间大大减少。

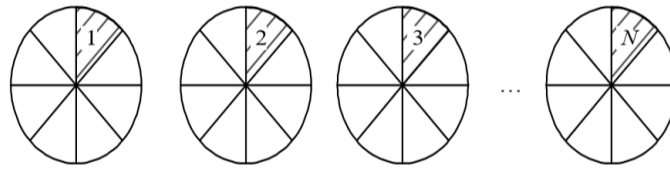


图 5-29 磁盘并行交叉存取方式

预测：

1. 对文件逻辑结构提出的要求：能提高检索速度、便于修改、降低文件的存储费用。