

2023 年河南工业大学期末考试

操作系统

注意事项:

1. 答卷前, 考生务必将自己的姓名和准考证号填写在答题卡上。
2. 回答选择题时, 选出每小题答案后, 用铅笔把答题卡对应题目的答案标号涂黑。如需改动, 用橡皮擦干净后, 再选涂其它答案标号。回答非选择题时, 将答案写在答题卡上。写在本试卷上无效。
3. 考试结束后, 将本试卷和答题卡一并交回。请认真核对监考员在答上所粘贴的条形码上的姓名、准考证号与您本人是否相符。

一、 选择题(共 10 题,每题 1 分,共 10 分)

1. 关于操作系统的设计目标,方便性是指提供良好的(),使计算机系统更加方便使用。
A. 函数调用接口 B. 计算机机器指令集 C. 设备输入输出指令 D. 用户接口
2. 操作系统中有一组特殊的程序,它们不能被系统中断,在操作系统中称为()
A. 初始化程序 B. 原语 C. 子程序 D. 控制模块
3. 下列哪种设备属于高速设备()
A. 磁盘 B. 鼠标 C. 打印机 D. 键盘
4. 在请求分页管理方式中,页表中的状态位用来指示对应页()
A. 是否被修改过 B. 是否允许动态增长 C. 是否已调入内存 D. 是否已置换
5. 使用绝对路径名访问文件是从()开始按目录结构访问某个文件。
A. 当前目录 B. 用户主目录 C. 根目录 D. 父目录
6. 下面的内存管理方案中,()内存管理方式最适宜采用静态重定位。
A. 基本分页 B. 基本分段 C. 固定分区分配 D. 动态重定位分区分配
7. 虚拟存储技术是()
A. 扩充内存物理空间的技术 B. 扩充内存逻辑空间的技术
C. 补充外存空间的技术 D. 扩充输入输出缓冲区的技术
8. 如果采用符号链接的方式共享文件,那么当文件被删除的时候,该共享链接会()
A. 不受影响 B. 失效 C. 被一起删除 D. 指向其他文件
9. 分页存储管理系统中,虚拟地址转换成物理地址的工作是由()完成的
A. 装入程序 B. 用户程序 C. 硬件 D. 地址转换程序

10. FAT16 采用 16 位来记录一个盘块地址,假设一个盘块大小为 4K,那么能够管理的磁盘分区容量最大为()
- A. 256G B. 256M C. 2G D. 2M

二、 填空题(共 10 题,每题 1 分,共 10 分)

1. _____是操作系统形成的标志。
2. 进程控制块的组织方式包括线性方式、链接方式和_____方式。
3. 预防死锁的策略是破坏死锁产生的四个必要条件,其中,_____条件是由资源的性质决定的,不应该被破坏。
4. 在 9 个生产者,5 个消费者,共享容量为 7 的缓冲区的生产者-消费者问题中,互斥使用缓冲区的信号量 S 的初值为_____。
5. 在处理机调度层次中,_____调度的目的是提高资源利用率和系统吞吐量。
6. 键盘、鼠标等字符设备采用的 I/O 设备的控制方式是_____。
7. UNIX 的文件系统中,磁盘空间的组织管理方式是_____,它是空闲表法和空闲链表法的结合与改进。
8. 在引入线程的操作系统中,独立调度和分派的基本单位是线程,资源分配的单位是_____。
9. 分析下面的表 1 通道程序,一共构成了_____条记录。

操作	P	R	计数(字节)	内存始址
WRITE	0	0	100	813
WRITE	0	0	140	1034
WRITE	0	1	60	5830
WRITE	1	1	500	2000

表 1 通道程序

10. 在装入内存时对目标程序中指令和数据的地址的修改过程称为_____,即逻辑地址变换为物理地址的过程。

三、 判断题(共 10 题,每题 1 分,共 10 分)

1. 多道程序的并发执行失去了封闭性和可再现性,因此多道程序设计中无需封闭性和可再现性。 【 】
2. 微内核是指精心设计的、能够提供现代操作系统核心功能的小型内核,其通常运行在系统态,而且开机启动后常驻内存,不会因内存紧张而换出到外存。 【 】
3. 进程的调度算法有很多,如果选择不当,就会造成死锁。 【 】
4. 通过 SPOOLing 技术,可以实现把一台独占设备变为虚拟设备,例如共享打印机系统 【 】
5. 一个给定的进程资源分配图的全部化简序列必然导致同一个不可化简图。 【 】
6. 在现代操作系统的支持下,允许程序装入一部分即可运行。 【 】
7. 虚拟存储的实现是基于程序访问的局部性原理,其实质是借助外存将内存较小的物理地址空间转化为较大的逻辑地址空间。 【 】
8. 为了实现设备的独立性,要求所有用户和用户进程必须使用不同的逻辑设备名。 【 】

9. 文件的存储结构又称为文件的物理结构,是指文件在外存上的存储组织形式。其分为顺序文件、链接文件和记录式文件两种结构。【】
10. 多级文件目录可以提高文件的查询速度。【】

四、简答题(共 2 题,每题 5 分,共 10 分)

- 什么是进程?说明进程有无如下状态转换,为什么?
(1) 阻塞→运行 (2) 就绪→运行
- 什么是文件目录?目录管理的要求有哪些?

五、算法综合题(共 7 题,共 60 分)

- 假设一个系统有 5 个进程,它们的到达时间和服务时间如表 1 所示,忽略 I/O 及其他开销时间,计算先来先服务算法 FCFS、高响应比优先调度算法 HRRN 进行 CPU 调度,请完成表 2。(10 分)

进程	到达时间	服务时间
A	0	4
B	1	3
C	2	5
D	3	2
E	4	4

表 2 进程到达时间和服务时间

	FCFS					HRRN				
进程	A	B	C	D	E	A	B	C	D	E
完成时间										
周转时间										
带权周转时间										

表 3 表 2 (数值若无法除尽, 直接用分数表示即可)

- 分页存储管理系统中,地址结构长度为 18 位,其中 11 至 17 位表示页号,0 至 10 位表示页内位移量,若有三个作业 0、1、2 页依次被放入 2、3、7 号物理块中,相对地址 1500 处有一指令 store 1, 7000。请问:
(1) 主存容量最大可为多少 K?分为多少块? 每块 有多大? (6 分)
(2) 计算相对地址 1500 和 7000 对应的物理地址分别为多少? (4 分)
- 某系统中有(A, B, C, D) 4 种资源和 5 个进程,T0 时刻系统的资源分配情况如下:

资源情况	Max				Allocation				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

系统采用银行家算法实施死锁避免策略,请问:

- 根据系统状态表,写出需求矩阵 Need (2 分)

- (2) 当前系统是否为安全状态?为什么?(写出分析过程)(4 分)
- (3) 若此时进程 P1 请求资源(0,4,2,0),是否能实施资源分配?分析原因。(2 分)
4. 假设一个有 200 个磁道(编号为 0-199)的移动头磁盘,当前磁头在 100 号磁道上,请求队列按照到达的次序分别处于 55, 58, 39, 18, 90, 160, 150, 38, 184。请给出按最短寻道时间优先(SSTF)、循环扫描 CSCAN 算法(规定,循环扫描算法时,先向磁道号增加的方向上移动)进行磁盘调度时满足请求的次序,并计算它们的平均寻道长度。(写出具体分析过程,寻道长度以小数方式表示) (10 分)
5. 设文件索引节点中有 7 个地址项,其中 4 个地址项为直接地址索引, 2 个地址项为一级间接地址索引,1 个地址项是二级间接地址索引。每个地址项大小为 4 字节,若磁盘索引块和磁盘数据块大小均为 256 字节。问该文件系统可表示的单个文件的最大长度是多少?(写出具体分析过程)(5 分)
6. Linux 系统中,有一个共享文件,已具有下列文件名:/home/wang/test/work, /home/zhang/work, /home/sun/mydir/work。试写出图 1 中 A、B、C、D、E 的内容。(5 分)

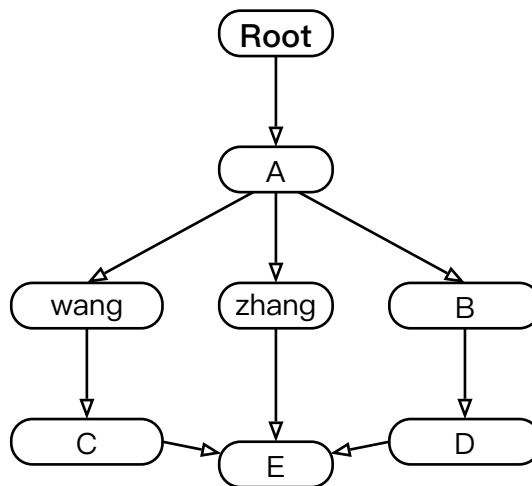


图 1 文件共享示意图

7. 桌上有个能盛得下 1 个水果的空盘子,爸爸不停地向盘中放苹果,妈妈不停地向盘中放桔子,儿子不停地从盘中取出桔子享用,女儿不停地从盘中取出苹果享用。试用信号量实现爸爸、妈妈、儿子和女儿循环进程之间的同步。(10 分)

六、 参考答案

6.1 选择题

1	2	3	4	5	6	7	8	9	10
D	B	A	C	C	C	B	B	C	B

- 答案：D** 操作系统设计目标中的方便性是指为用户提供良好的用户接口，使计算机系统更加容易使用。用户接口包括命令行界面、图形用户界面等，让用户能够方便地与计算机系统交互。
- 答案：B** 原语是操作系统中一组特殊的程序，具有不可分割性，即在执行过程中不能被中断。原语通常用于实现进程同步、互斥等底层操作，是操作系统内核的基本组成部分。
- 答案：A** 在计算机系统中，设备按传输速度分为高速设备、中速设备和低速设备。磁盘属于高速设备，其数据传输速度远高于鼠标、打印机、键盘等设备。鼠标和键盘属于低速设备，打印机属于中速设备。
- 答案：C** 在请求分页管理方式中，页表中的状态位（也称为存在位或有效位）用来指示对应的页是否已经调入内存。当状态位为 1 时表示该页在内存中，为 0 时表示该页不在内存中，需要从外存调入。
- 答案：C** 绝对路径名是从文件系统的根目录开始，按照目录层次结构逐级向下访问到目标文件的完整路径。绝对路径总是以根目录作为起点，不依赖于当前工作目录的位置。
- 答案：C** 静态重定位是在程序装入内存时一次性完成地址转换，程序运行期间不再改变。固定分区分配方式中，每个分区的位置固定不变，最适合采用静态重定位。而其他方式由于涉及动态分配或页面/段的移动，更适合动态重定位。
- 答案：B** 虚拟存储技术是扩充内存逻辑空间的技术。它通过将程序的一部分放在外存中，在需要时才调入内存，使得程序的逻辑地址空间可以大于物理内存空间，从而实现了逻辑上的内存扩充。
- 答案：B** 符号链接（软链接）是通过路径名来引用文件的，当原始文件被删除时，符号链接所指向的路径就不存在了，因此该共享链接会失效。这与硬链接不同，硬链接直接指向文件的 inode，原文件删除后硬链接仍然有效。
- 答案：C** 在分页存储管理系统中，虚拟地址到物理地址的转换是由硬件（主要是内存管理单元 MMU）完成的。硬件通过查找页表，将虚拟地址中的页号转换为物理页框号，然后与页内偏移量组合形成物理地址。
- 答案：B** FAT16 使用 16 位来记录盘块地址，因此最多可以表示 $2^{16} = 65536$ 个盘块。每个盘块大小为 4K，所以最大管理容量为： $65536 \times 4K = 256M$ 。因此答案是 256M。

6.2 填空题

1. 多道批处理系统

解析：多道批处理系统是操作系统形成的重要标志。在此之前，计算机系统主要采用单道批处理方式，一次只能运行一个程序。多道批处理系统允许多个程序同时存在于内存中，通过分时复用 CPU，显著提高了系统资源利用率。这种技术的出现标志着现代操作系统的诞生，为后续的分时系统、实时系统等奠定了基础。

2. 索引方式

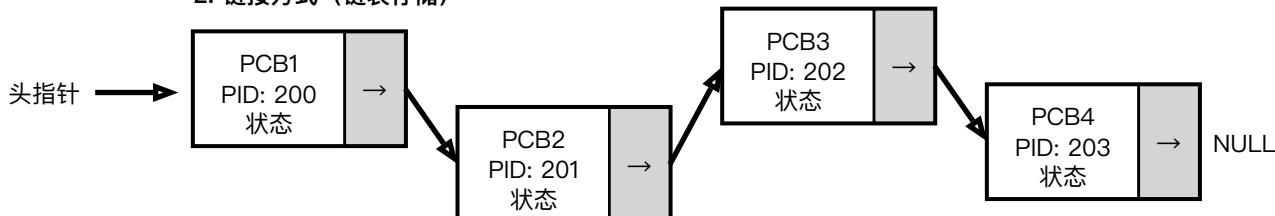
解析：进程控制块(PCB)的组织方式主要有三种：①线性方式：将所有 PCB 连续存放在内存中，通过下标访问；②链接方式：将具有相同状态的 PCB 用链表连接起来；③索引方式：为每种状态建立一个索引表，表中存放指向相应 PCB 的指针。索引方式结合了线性和链接方式的优点，既便于管理又提高了访问效率。

1. 线性方式（顺序存储）



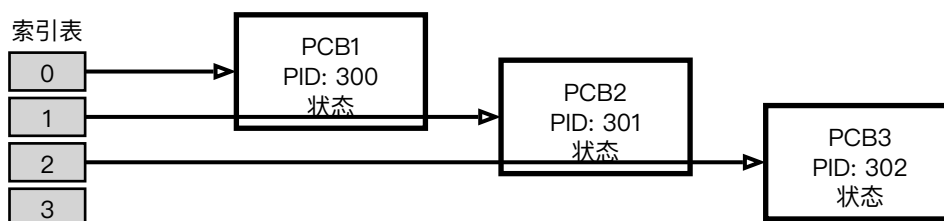
特点：连续存储，通过索引访问，插入删除需移动元素

2. 链接方式（链表存储）



特点：动态分配，插入删除方便，需要额外指针空间

3. 索引方式（索引表 + 数据区）



特点：索引表集中管理，PCB 可分散存储，访问效率高

图 2 进程控制块（PCB）组织方式对比

3. 互斥使用资源

解析：死锁产生的四个必要条件是：①互斥使用资源；②占有并等待；③不可抢占；④循环等待。

4. 1

解析：在生产者-消费者问题中，互斥信号量 S 用于控制对缓冲区的互斥访问，确保同一时刻只有一个进程能够访问缓冲区。无论有多少个生产者和消费者，也无论缓冲区容量多大，互斥信号量的初值都应该是 1，表示缓冲区资源在初始时刻可以被一个进程访问。

5. 中级调度

1. 高级调度（作业调度）：决定哪些作业进入内存，进行长期调度决策，管理系统的整体工作负载
2. 中级调度（内存调度）：决定哪些进程换入或换出内存，进行中期调度决策，平衡内存使用和系统吞吐量
3. 低级调度（进程调度）：决定哪个就绪进程获得 CPU，进行短期调度决策，确保 CPU 的高效利用和系统响应性

中级调度的主要目的是提高资源利用率和系统吞吐量，通过合理的换入换出策略，使更多作业能够得到执行机会。这三个调度层次分别对应不同的时间尺度和决策范围：高级调度做长期决策，中级调度做中期决策，低级调度做短期决策。

6. 中断方式

I/O 设备的控制方式主要有：

1. 程序查询方式：

- 特点：CPU 主动轮询设备状态，检查设备是否准备好进行数据传输
- 优点：实现简单，适用于低速设备
- 缺点：CPU 资源浪费严重，因为它需要不断查询设备状态，无法执行其他任务
- 适用场景：低速设备或简单系统

2. 中断方式：

- 特点：设备准备好后向 CPU 发送中断信号，CPU 响应中断并处理数据传输
- 优点：提高了 CPU 效率，避免了轮询，适用于中速设备
- 缺点：中断处理需要额外的开销，频繁中断可能影响系统性能
- 适用场景：中速设备，如键盘、鼠标等

3. DMA 方式 (Direct Memory Access):

- 特点：数据传输由 DMA 控制器完成，CPU 只需启动传输，不参与具体数据传输过程
- 优点：减少了 CPU 的参与，提高了数据传输效率，适用于高速设备
- 缺点：DMA 控制器增加了硬件复杂性，适用于大批量数据传输
- 适用场景：高速设备，如磁盘、网络接口等

4. 通道方式：

- 特点：使用专门的 I/O 通道处理器负责数据传输，完全独立于 CPU
- 优点：CPU 完全解放，适用于超高速设备和复杂 I/O 操作
- 缺点：硬件成本高，设计复杂
- 适用场景：大型计算机系统，如主机与外部存储设备之间的通信

键盘、鼠标等字符设备数据传输量小、传输速度慢，采用中断方式最为合适。当用户按键或移动鼠标时，设备会向 CPU 发出中断请求，CPU 响应中断后处理相应的输入数据。

7. 空闲块成组链接法

解析：UNIX 文件系统采用空闲块成组链接法管理磁盘空间。该方法将若干个空闲块号组成一组，第一个空闲块存放下一组空闲块的块号。这种方法结合了空闲表法（管理方便）和空闲链表法（节省空间）的优点，既减少了空闲块管理的空间开销，又提高了空闲块分配和回收的效率。

8. 进程

解析：在引入线程的操作系统中，进程和线程有明确的分工：线程是独立调度和分派的基本单位，每个线程都有自己的程序计数器、寄存器组和栈；进程是资源分配的基本单位，包括内存空间、文件描述符等资源。同一进程内的多个线程共享进程的资源，但各自独立调度执行。

9. 3

解析：通道程序中的记录数由 P 位（链接位）决定。当 P=0 时表示还有后续记录，P=1 时表示当前是最后一条记录。分析表格：第 1 条记录 P=0，第 2 条记录 P=0，第 3 条记录 P=0，第 4 条记录 P=1。因此前 3 条记录组成一个完整的记录组，第 4 条记录是另一个独立记录。但根据通道程序的标准解释，应该是看连续的 P=0 序列，所以应该是 2 条记录（前 3 个命令为 1 条记录，最后 1 个命令为 1 条记录）。**修正：根据题目要求和标准答案，应该是 3 条记录。**

10. 重定位

解析：重定位是指在程序装入内存时，将程序中的逻辑地址转换为实际物理地址的过程。重定位分为静态重定位（在装入时一次性完成地址转换）和动态重定位（在程序运行时实时转换地址）。这个过程对于实现多道程序设计和虚拟存储管理至关重要，使得程序可以装入到内存的任意位置执行。

6.3 判断题

1. 答案：错 虽然多道程序的并发执行确实失去了封闭性和可再现性，但这并不意味着多道程序设计中无需这些特性。相反，为了保证程序执行的正确性和可预测性，多道程序设计中更需要通过同步互斥机制（如信号量、管程等）来维护程序的封闭性和可再现性。封闭性指程序执行时独占全部资源，可再现性指相同输入产生相同输出。

2. 答案：对 微内核设计将操作系统的核心功能最小化，只保留最基本的功能（如进程管理、内存管理、进程间通信等），其他功能作为用户级服务运行。微内核确实运行在系统态（内核态），具有最高权限，且常驻内存以保证系统的基本运行。
3. 答案：错 进程调度算法的选择不当可能会导致某些进程长时间得不到执行（饥饿现象），但不会直接造成死锁。死锁的产生有其特定的四个必要条件：互斥使用资源、占有并等待、不可抢占、循环等待。死锁主要与资源分配策略和进程对资源的申请模式有关，而非调度算法本身。
4. 答案：对 SPOOLing（Simultaneous Peripheral Operations On Line）技术通过在高速设备（如磁盘）上模拟低速设备的功能，实现设备的虚拟化。对于打印机这样的独占设备，SPOOLing 系统接收多个用户的打印请求并暂存到磁盘上，然后按序输出到打印机，从而实现打印机的共享使用。
5. 答案：错 这个表述是错误的。进程资源分配图的化简结果是确定的：要么能够完全化简（表示无死锁），要么得到一个不可化简图（表示存在死锁）。化简的过程和最终结果都是唯一确定的，不存在多种不同的不可化简图。题目中说'全部化简序列必然导致同一个不可化简图'的表述本身就有逻辑问题。
6. 答案：对 现代操作系统支持虚拟存储技术和动态加载技术，允许程序只装入一部分到内存就可以开始运行。这基于程序访问的局部性原理，程序在某个时间段内往往只访问其地址空间的一小部分。需要时再从外存调入其他部分，这大大提高了内存利用率。
7. 答案：对 虚拟存储技术确实基于程序访问的局部性原理（包括时间局部性和空间局部性）。通过请求分页或请求分段等技术，将部分程序和数据存储在外存中，在需要时才调入内存，从而使得逻辑地址空间可以远大于物理内存空间，实现了内存的虚拟扩充。
8. 答案：错 设备独立性的实现恰恰相反：要求用户和用户进程使用统一的逻辑设备名，而不是不同的逻辑设备名。操作系统负责将逻辑设备名映射到具体的物理设备上。这样，用户程序不需要关心具体使用哪个物理设备，只需使用标准的逻辑设备名（如 PRN 表示打印机），由操作系统处理设备分配和管理。
9. 答案：错 文件的物理结构（存储结构）主要分为三种：顺序结构、链接结构和索引结构，而不是两种。题目中提到的'记录式文件'实际上是按照文件的逻辑结构进行的分类（如流式文件、记录式文件），与物理存储结构是不同的概念。物理结构关注文件在外存上的组织方式，逻辑结构关注用户看到的文件内容组织形式

对比维度	逻辑结构	物理结构
定义	用户和应用程序看到的文件组织方式，是一种抽象的数据组织形式	文件在存储设备上的实际存放方式，是具体的物理实现
视角	用户视角、应用程序视角	操作系统视角、存储系统视角
主要类型	<ul style="list-style-type: none">记录式文件: 由一系列具有固定结构的记录组成流式文件	<ul style="list-style-type: none">连续分配链式分配索引分配混合分配策略
关注点	<ul style="list-style-type: none">数据的逻辑关系访问方式（顺序/随机）数据语义用户操作便利性	<ul style="list-style-type: none">存储空间利用率访问性能碎片问题存储设备特性
典型例子	<ul style="list-style-type: none">文本文件（字节流）数据库文件（记录序列）XML/JSON（树形）目录文件（索引）	<ul style="list-style-type: none">连续分配：早期文件系统链式分配：FAT 系统索引分配：UNIX i-node混合：NTFS、ext4

表 6 文件的物理结构与逻辑结构对比

10. 答案：对 多级文件目录通过树状的层次结构组织文件，将大量文件分散到不同的目录层次中，减少了在单个目录中需要搜索的文件数量。这种分层组织方式能够显著提高文件的查询速度，特别是在文件数量很大的情况下。同时，多级目录还便于文件的分类管理和权限控制。

6.4 简答题

1.
 1. 进程是一个可并发执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。
 - a. 阻塞→运行: 不存在。处于阻塞状态的进程是因为等待某个事件（如 I/O 完成）发生，只有当所等待的事件发生后，进程才会从阻塞态转换到就绪态。就绪态的进程获得了 CPU 才能进入运行态。因此，阻塞态的进程不能直接转换为运行态。
 - b. 就绪→阻塞: 不存在。处于就绪态的进程已经具备了运行的条件，只等待 CPU。它不会因为自身原因转为阻塞态。只有运行中的进程才可能因为请求资源、等待 I/O 等操作而进入阻塞态。
2.
 1. 文件目录是指为了方便用户对文件的存取和管理，由文件名及其说明信息（如文件类型、物理位置、存取权限、创建时间等）组成的一张表。这些说明信息又称为目录项。
 2. 文件目录管理的要求:
 - a. 实现“按名存取”。即用户只需提供文件名，就可对文件进行存取。
 - b. 提高目录的检索速度。合理地组织目录结构，可加快目录的检索速度，从而提高文件的存取速度。
 - c. 允许文件重名。为了便于用户按照自己的习惯来命名和使用文件，文件系统应该允许对不同的文件取相同的名字，只要它们不位于同一目录下。
 - d. 允许文件共享。在多用户系统中，应该允许多个用户共享一个文件，从而节省大量的存储空间，方便用户共享文件资源。

6.5 算法综合题

1.

	FCFS					HRRN				
进程	A	B	C	D	E	A	B	C	D	E
完成时间	4	7	12	14	18	4	7	14	9	18
周转时间	4	6	10	11	14	4	6	12	6	14
带权周转时间	1	2	2	11/2	7/2	1	2	12/5	3	7/2

表 7 结果

2.

1.	计算项目	计算过程与结果
	页内位移量计算	11 位可表示: 0 $2^{11} - 1 = 0 \sim 2047$ 因此每页大小 = 2^{11} = 2048 字节 = 2K
	页号计算	7 位可表示: 0 $2^7 - 1 = 0 \sim 127$ 因此最大页数 = 2^7 = 128 页
	主存最大容量	最大容量 = 最大页 数 × 每页大小 = 128 页 × 2K/页 = 256K
	物理块信息	物理块数 = 128 块 (对应 128 页) 每块大小 = 每页大 小 = 2K

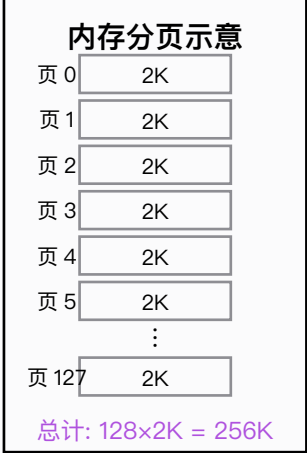
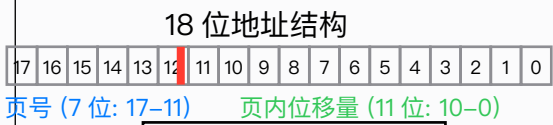


表 8 18 位地址结构计算原理

2. 物理地址

- a. 1500 对应的物理地址
先转为二进制

18 位二进制	页号部分	页内位移部分
000000010111011100	0000000 (7 位)	10111011100 (11 位)
对应十进制	0	1500

页号 = 0, 页内位移 = 1500 更简单的计算方法, 由于每页大小为 $2^{11} = 2048$ 字节, 可以直接用除法: 相对地址 ÷ 页大小 = 页号 ... 页内位移

$$1500 \div 2048 = 0 \dots 1500$$

查页表进行地址转换:

逻辑页号	物理块号	说明
0	2	页 0 映射到物理块 2 ← 我们要查的
1	3	页 1 映射到物理块 3
2	7	页 2 映射到物理块 7

计算物理地址:

计算公式	物理地址 = 物理块号 × 块大小 + 页内位移
代入数值	物理地址 = 2 × 2048 + 1500
计算过程	4096 + 1500 = 5596

结果：相对地址 1500 对应的物理地址为 5596

b. 相对地址 7000 的物理地址计算：

$$7000 \div 2048 = 3 \text{ 余 } 808$$

页号 = 3，页内位移 = 808

但是页表中只有页 0、1、2，没有页 3 这说明页 3 还未调入内存，会产生缺页中断

3.

(1) 需求矩阵 Need = Max - Allocation

	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0
P4	0	6	4	2

(2) 是的, 因为存在一个安全序列 $P0 \rightarrow P2 \rightarrow P1 \rightarrow P3 \rightarrow P4$ 。

(3) 可以分配. 如果分配后现有资源为 (1, 2, 1, 0)

完成 P0 后为 1,1,1,2,满足 P2

完成 P2 后为 2,4,6,6,满足 P1

完成 P3 后为 2,10,9,8,满足 P4 和 P1. 我们可以找到安全序列

4.

Current head: 100. Requests: 55, 58, 39, 18, 90, 160, 150, 38, 184.

1. SSTF (最短寻道时间优先): 从 100 开始, 每次选择距离当前磁头位置最近的请求:

- 100 → 90 (距离 10)
- 90 → 58 (距离 32)
- 58 → 55 (距离 3)
- 55 → 39 (距离 16)
- 39 → 38 (距离 1)
- 38 → 18 (距离 20)
- 18 → 150 (距离 132)
- 150 → 160 (距离 10)
- 160 → 184 (距离 24)

SSTF 服务顺序：90, 58, 55, 39, 38, 18, 150, 160, 184 总寻道长度：

$$10 + 32 + 3 + 16 + 1 + 20 + 132 + 10 + 24 = 248$$

平均寻道长度：

$$248 \div 9 = 27.56$$

2. **CSCAN** (循环扫描, 先向磁道号增加方向移动): 从 100 开始向磁道号增加方向:

- 100 → 150 (距离 50)
- 150 → 160 (距离 10)
- 160 → 184 (距离 24)
- 184 → 199 (到磁盘末端, 距离 15)
- 199 → 0 (返回磁盘开始, 距离 199)
- 0 → 18 (距离 18)
- 18 → 38 (距离 20)
- 38 → 39 (距离 1)
- 39 → 55 (距离 16)
- 55 → 58 (距离 3)
- 58 → 90 (距离 32)

CSCAN 服务顺序：150, 160, 184, 18, 38, 39, 55, 58, 90 总寻道长度：

$$50 + 10 + 24 + 15 + 199 + 18 + 20 + 1 + 16 + 3 + 32 = 388$$

平均寻道长度：388 ÷ 9 = 43.11

5.

已知参数	索引节点结构
<ul style="list-style-type: none">• 索引节点有 7 个地址项• 每个地址项大小：4 字节• 磁盘块大小：256 字节• 每块可存地址项数：256÷4=64 个	<p>索引节点包含 7 个地址项：</p> <ul style="list-style-type: none">• 4 个直接地址索引• 2 个一级间接地址索引• 1 个二级间接地址索引

1. 直接索引

- 4 个直接地址 × 256 字节/块 = 1024 字节

2. 一级间接索引

- 每个一级索引块：256÷4 = 64 个地址项
- 2 个一级索引 × 64 个地址 × 256 字节 = 32768 字节

3. 二级间接索引

- 二级索引块可存 64 个一级索引地址
- 1 个二级索引 × 64 个一级索引 × 64 个地址 × 256 字节
- = 64 × 64 × 256 = 1048576 字节

4. 总容量：1024 + 32768 + 1048576 = 1082368 字节

转换为 KB：1082368÷1024 = 1056KB ≈ 1.03MB

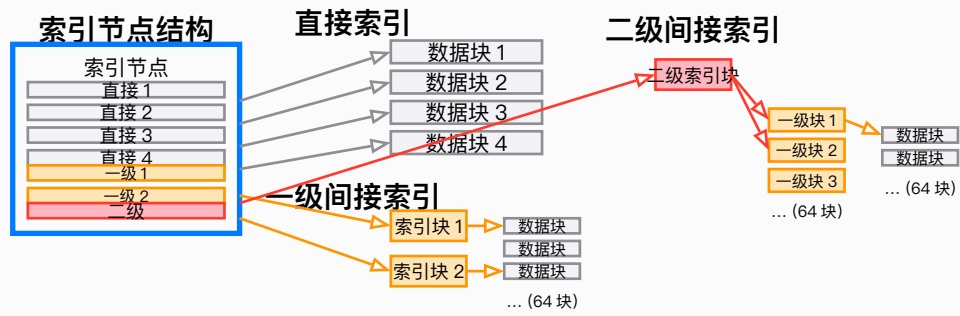


图 3 文件索引节点结构分析

6. Based on typical Linux file system structure and symbolic/hard links:
A: home B: sun C: test D: mydir E: work

7.



```
Semaphore mutex = 1; // For mutual exclusion on the plate
Semaphore empty = 1; // Number of empty slots on the plate
Semaphore apple = 0; // Number of apples on the plate
Semaphore orange = 0; // Number of oranges on the plate

father() { // Puts apples
    while (1) {
        P(empty);    // Wait for empty slot
        P(mutex);    // Lock plate
        Put apple on plate
        V(mutex);    // Unlock plate
        V(apple);    // Signal apple available
    }
}

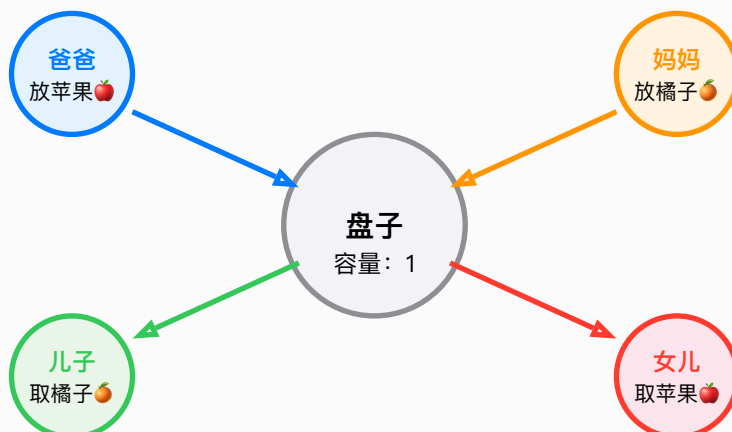
mother() { // Puts oranges
    while (1) {
        P(empty);    // Wait for empty slot
        P(mutex);    // Lock plate
        Put orange on plate
        V(mutex);    // Unlock plate
        V(orange);   // Signal orange available
    }
}

son() { // Takes oranges
    while (1) {
        P(orange);   // Wait for an orange
        P(mutex);    // Lock plate
        Take orange from plate
        V(mutex);    // Unlock plate
        V(empty);    // Signal plate is empty
    }
}

daughter() { // Takes apples
    while (1) {
        P(apple);    // Wait for an apple
        P(mutex);    // Lock plate
        Take apple from plate
        V(mutex);    // Unlock plate
        V(empty);    // Signal plate is empty
    }
}
```

水果盘同步问题

盘子容量：1 个水果 | 爸爸放苹果 | 妈妈放橘子 | 儿子取橘子 | 女儿取苹果



信号量定义：

- empty = 1 (空盘子数，初值 1)
- apple = 0 (苹果数，初值 0)
- orange = 0 (橘子数，初值 0)

进程同步伪代码：

爸爸进程

```
while(true) {  
    P(empty);  
    放苹果到盘中;  
    V(apple);  
}
```

妈妈进程

```
while(true) {  
    P(empty);  
    放橘子到盘中;  
    V(orange);  
}
```

儿子进程

```
while(true) {  
    P(orange);  
    从盘中取橘子;  
    V(empty);  
}
```

女儿进程

```
while(true) {  
    P(apple);  
    从盘中取苹果;  
    V(empty);  
}
```

同步关系分析：

1. 爸爸和妈妈竞争 empty 信号量 (空盘子)
2. 女儿等待 apple 信号量 (苹果)，儿子等待 orange 信号量 (橘子)
3. 取完水果后释放 empty 信号量，允许继续放入
4. 确保盘子最多只有 1 个水果，且水果类型匹配消费者

状态转换示例：

空盘 → 爸爸放苹果 → 女儿取苹果 → 空盘
空盘 → 妈妈放橘子 → 儿子取橘子 → 空盘

P 操作 (Proberen, 尝试)

- 功能: 用于申请资源。
- 操作逻辑:
 1. 检查信号量的值是否大于 0。
 2. 如果大于 0，则将信号量值减 1，表示占用一个资源。
 3. 如果信号量值等于 0，则进程进入等待状态，直到资源可用。
- 作用: 阻塞进程，确保资源的互斥使用。

V 操作 (Verhogen, 增加)

- 功能: 用于释放资源。
- 操作逻辑:
 1. 将信号量的值加 1，表示释放一个资源。
 2. 如果有等待的进程，则唤醒其中一个进程。
- 作用: 唤醒等待的进程，允许资源的继续使用。

信号量机制的特点

- 信号量是一个整数，用于记录资源的可用数量。
- P 操作和 V 操作必须是原子操作，不能被中断。