

2019 至 2020 学年 第 2 学期

《操作系统原理》 试卷 A 卷

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

一、 选择题 (共 15 题, 每题 1 分, 共 15 分) 答案填入答题纸的表格中

1. 页式存储管理中的地址格式是()地址。
A. 线性 B. 一维 C. 二维 D. 三维
2. 成组链接法可用于()。
A. 磁盘空间的管理 B. 磁盘的驱动调度
C. 文件目录的查找 D. 页式虚拟存贮管理中的页面调度
3. 通道是一种()。
A. I/O 端口 B. 数据通道 C. 软件工具 D. I/O 专用处理器
4. 任何两个并发进程之间()。
A. 一定存在互斥关系 B. 一定存在同步关系
C. 一定彼此独立无关 D. 可能存在同步或互斥关系
5. 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
6. 可变式分区分配方案中, 某一作业完成后, 系统收回其主存空间, 并与相邻空闲区合并, 为此需修改空闲区表, 造成空闲区数减 1 的情况是()。
A. 无上邻空闲区, 也无下邻空闲区 B. 有上邻空闲区, 但无下邻空闲区
C. 有下邻空闲区, 但无上邻空闲区 D. 有上邻空闲区, 也有下邻空闲区
7. 一种既有利于短作业又兼顾长期作业的作业调度方式是()。
A. 先来先服务 B. 均衡调度 C. 最短作业优先 D. 多级反馈队列调度
8. 进程从运行状态进入就绪状态的原因可能是()。
A. 被选中占有处理机 B. 等待某一事件 C. 等待的事件已发生 D. 时间片用完
9. 文件系统中, 目录管理的最基本功能是()。
A. 实现虚拟存储 B. 实现文件的按名存取 C. 提高外存的读写速度 D. 用于存储系统文件
10. 静态重定位是在作业()中进行的。
A. 编译过程 B. 链接过程 C. 装入过程 D. 执行过程
11. 一个进程从提交系统到完成的时间间隔称为()。
A. 等待时间 B. 周转时间 C. 响应时间 D. 完成时间

12. 当处理机没有将时间用在处理指令, 而都用在频繁地将进程换入换出时, 这种情况称为()。
- A. 页面置换 B. 局部性原理 C. 抖动 D. 对换
13. 磁盘调度主要是为了优化()。
- A. 寻道时间 B. 旋转延迟时间 C. 传输时间 D. 寻找磁盘的时间
14. UNIX 文件系统中的文件索引节点包含 13 个地址项, 其中 i.addr(0) i.addr(9)是直接地址, i.addr(10)是一级间接索引, i.addr(11)是二级间接索引, i.addr(12)是三级间接索引。假设盘块大小为 4K, 那么各个索引块也采用 i.addr(0) i.addr(10)这 11 个地址项能够管理的最大文件是()。
- A. 40K B. 4M C. 40K+4M D. 40K+4M+4G
15. 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)

- 在分时系统中, 假设用户数目为 50 时, 为保证响应时间不超过 2s, 此时系统的时间片最大应为 _____ ms。
- 有两个程序 A、B, 它们使用同一个设备进行 I/O 操作, 并按 A、B 的优先次序执行, 每个程序的计算和 I/O 操作时间如表 1 所示。假设调度时间可忽略不计, 请计算多道环境下它们的总运行时间是 _____ ms。

	程序	A	B
操作			
	计算	30	60
	I/O	40	30
	计算	10	10

表 1 表 1 程序运行时间表 (单位 ms)

- UNIX 系统采用的外存分配方式为 _____, 索引结点中共设了 13 个地址项。
- 用信号量 S 实现对系统中 4 台打印机的互斥使用, 若 S.value 的当前值为-2, 表示 S.L 队列中有 _____ 个等待进程。
- 消息缓冲队列通信机制是通过内存中公用的 _____ 进行进程通信。
- 已知某分页系统, 主存容量为 640K, 页面大小为 1K, 对一个 4 页大的作业, 第 0、1、2、3 页被分配到内存的 2、6、9、7 块中。求: 将十进制的逻辑地址 2985 转换成物理地址是 _____。
- 硬盘采用的 I/O 设备的控制方式是 _____ 方式。
- 假设盘块的大小为 1KB, 硬盘的大小为 500MB, 采用显式链接分配时, 其 FAT 表项至少需要 _____ 个二进制位。
- 如果利用 30 行、16 列的位示图来表示空闲盘块的状态, 假定行号、列号和盘块号均从 1 开始编号, 则在进行盘块分配时, 若第一次找到的空闲盘块 (即该位的值为“0”) 处于第 8 行、第 15 列, 则相应的盘块号为 _____。
- 预防死锁的策略是破坏死锁产生的必要条件, 其中哪个条件是由资源的性质决定的, 而不应该被破坏的? _____

三、简答题 (共 3 题, 每题 5 分, 共 15 分)

1. 进程与程序的区别?
2. 什么是虚拟存储器? 其具有哪些基本特征?
3. 什么是进程同步? 同步机制应遵循的原则是什么?

四、算法综合题 (共 6 题, 共 50 分)

1. 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 2 所示, 忽略 I/O 及其他开销时间, 计算先来先服务 FCFS 算法、高响应比优先 HRRN 进行 CPU 调度, 请完成表 3。(10 分)【注: 以小数方式表示】

进程	到达时间	服务时间
A	0	3
B	2	5
C	3	4
D	5	4
E	8	2

表 2 表 2 进程到达和服务时间

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

表 3 表 3 调度结果 【注: 以小数方式表示】

2. 系统中有(A, B, C, D)四种资源和五个进程。在银行家算法中, 某时刻出现下述资源分配情况时:

Process	Allocation (A,B,C,D)	Need (A,B,C,D)	Available (A,B,C,D)
P0	(0,0,3,2)	(0,0,1,2)	(1,6,2,2)
P1	(1,0,0,0)	(1,7,5,0)	
P2	(1,3,5,4)	(2,3,5,6)	
P3	(0,3,3,2)	(0,6,5,2)	
P4	(0,0,1,4)	(0,6,5,6)	

试问:

1. 该状态是否安全? 请写出分析过程。(5 分)
 2. 若进程 P2 提出请求 Request(1,2,2,2), 系统能否将资源分配给它? 请写出分析过程。(3 分)
 3. 如果系统立即满足 P2 的请求, 系统是否会立即进入死锁状态? 为什么?(2 分)
3. 在一个请求页式存储系统中, 一个程序的页面走向为 3,4,1,3,2,0,3,2,4,2,3,0,1,2, 并分别采用 FIFO、LRU 页面置换算法。假设分配给该程序的存储块数 M 为 4, 求出在访问过程中发生的缺页次数和缺页率。(写出具体的分析过程, 缺页率可保留两位小数)(10 分)
4. 假设有一个 200 个磁道 (编号为 0-199) 的移动头磁盘, 请求队列中是一些随机请求, 它们按照到达的次序分别处于 100、183、40、122、13、124、65、69、90、50。当前在 55 号磁道上。请给出按先来先服务 FCFS, 扫描 SCAN 算法 (规定, 扫描算法时, 先向磁道号减少的方向上移动) 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。(写出具体的分析过程, 寻道长度以小数方式表示)(10 分)
5. Linux 系统中, 有一个共享文件, 它具有下列文件名: /home/wang/test/work、/home/zhang/work、/home/sun/mydir/work。试写出图 1 中的 A、B、C、D、E 的内容。(5 分)

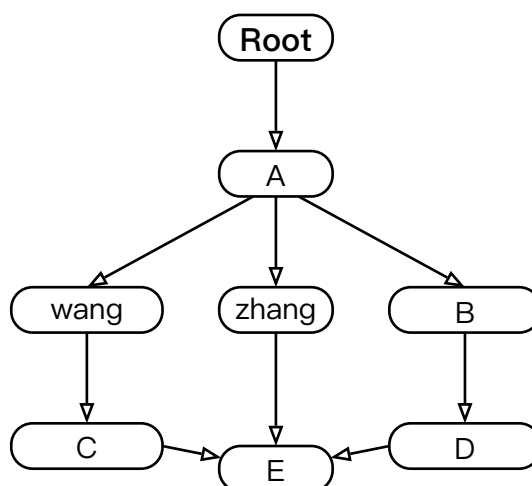


图 1 文件共享示意图

6. 在 MS-DOS 中有一文件 A, A 依次占用 3, 6, 8, 10, 5 五个盘块。请画出 MS-DOS 文件系统中, 文件 A 在各盘块间的链接情况及 FAT 的情况。(5 分) 答题空间 (For drawing)

五、程序设计题 (共 1 题, 每题 10 分, 共 10 分)

1. 假设有 3 个并发进程 P, Q, R, 其中 P 负责从输入设备上读入信息, 并传送给 Q, Q 将信息加工后传送给 R, R 负责打印输出。进程 P, Q 共享一个有 m 个缓冲区组成的缓冲池; 进程 Q, R 共享一个有 n 个缓冲区组成的缓冲池 (假设缓冲池足够大, 进程间每次传输信息的单位均小于等于缓冲区长度), 请用 PV 操作写出满足上述条件的并发程序。

六、参考答案与解析

6.1 选择题

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	A	D	D	C	D	D	D	B	C	B	C	A	D	B

第1题 答案：C

解析：本题考查页式存储管理的地址结构。页式存储管理中，逻辑地址由页号和页内偏移量两部分组成，是一种二维地址结构。选项A、B、D描述不准确。

第2题 答案：A

解析：本题考查磁盘空间管理方法。成组链接法是一种用于管理磁盘空闲块的高效方法，主要用于分配和回收磁盘空间。其他选项如驱动调度（如SCAN、CSCAN）、目录查找（如哈希、B树）、页面调度（如FIFO、LRU）是不同的操作系统功能。

第3题 答案：D

解析：本题考查通道的概念。通道是独立于CPU的专门负责数据输入输出传输控制的I/O专用处理器，它可以控制外部设备与内存之间的数据交换，从而解放CPU。

第4题 答案：D

解析：本题考查并发进程间的关系。并发进程在执行过程中可能会共享资源或需要合作完成任务，因此它们之间可能存在因共享资源而产生的互斥关系，或因合作而产生的同步关系。它们不一定总是互斥或同步，也可能在某些情况下彼此独立。

第5题 答案：C

解析：本题考查请求分页管理中的页表项。页表中的状态位（有效位/驻留位）用于指示该页当前是否已经调入内存。修改位指示页面是否被修改，访问位指示页面是否被访问，用于页面置换算法。

第6题 答案：D

解析：本题考查可变分区存储管理中的空闲区合并。当回收的作业区其上下都有相邻的空闲区时，这三个区（上邻空闲区、回收区、下邻空闲区）会合并成一个大的空闲区，使得总的空闲区数目减2后加1，即净减1。如果只有上邻或下邻，则数目不变。如果均无，则数目加1。

第7题 答案：D

解析：本题考查作业调度算法的特性。多级反馈队列调度算法通过设置多个不同优先级的队列，并结合时间片轮转，既能让短作业在较高优先级队列中快速完成，也能让长作业在较低优先级队列中得到服务，避免饿死，因此兼顾了短作业和长作业。

第8题 答案：D

解析：本题考查进程状态转换。进程从运行状态进入就绪状态，通常是因为分配给该进程的时间片已经用完，CPU被调度程序分配给其他进程，而该进程并未阻塞，仍具备运行条件。等待某一事件导致进入阻塞态；等待事件发生导致从阻塞态进入就绪态；被选中占有处理机导致从就绪态进入运行态。

第9题 答案：B

解析：本题考查文件系统的目录管理功能。目录管理的核心功能是组织和管理文件，提供文件名到文件物理存储位置的映射，即实现文件的“按名存取”。

第10题 答案：C

解析：本题考查重定位的方式。静态重定位是在作业装入内存时，由装入程序将指令和数据中的逻辑地址转换为物理地址，且装入后不能再移动。编译和链接过程处理的是符号地址和相对地址。

第11题 答案：B

解析：本题考查作业调度中的时间度量。周转时间是指从作业提交给系统开始，到作业完成为止的全部时间，包括等待调度时间、在就绪队列中等待CPU的时间、CPU执行时间以及I/O操作时间。

第12题 答案：C

解析：本题考查系统性能问题。抖动（Thrashing）是指在请求分页系统中，由于分配给进程的物理页面太少，导致进程运行频繁发生缺页中断，大部分时间都用于页面置换，而不是有效执行指令，使得系统效率急剧下降的现象。

第 13 题 答案：A

解析：本题考查磁盘调度的目标。磁盘 I/O 操作的时间主要由寻道时间、旋转延迟时间和数据传输时间组成。其中寻道时间是磁头移动到目标磁道所需的时间，通常是占比最大的部分，因此磁盘调度的主要目标是减少平均寻道时间。

第 14 题 答案：D

解析：本题考查 UNIX 文件系统的文件大小计算。直接地址：i.addr(0)~i.addr(9) 共 10 项，每项指向一个盘块，大小为 $10 \times 4K = 40K$ 。一级间接索引：i.addr(10) 指向一个索引块，该索引块大小为 4K。每个地址项假设为 4 字节，则一个索引块可存放 $\frac{4 \times 1024}{4} = 1024$ 个盘块地址。所以一级间接索引能管理 $1024 \times 4K = 4M$ 。二级间接索引：i.addr(11) 指向一个二级索引块，该块又指向 1024 个一级索引块，每个一级索引块再指向 1024 个数据块。所以二级间接索引能管理 $1024 \times 1024 \times 4K = 4G$ 。三级间接索引：i.addr(12) 同理能管理 $1024 \times 4G = 4T$ 。题目给的答案 D 是 $40K + 4M + 4G$ ，这意味着它考虑了 i.addr(0) 到 i.addr(11) 这 12 项（10 个直接，1 个一级间接，1 个二级间接）。

第 15 题 答案：B

解析：本题考查 FAT16 文件系统的容量计算。FAT16 使用 16 位记录盘块地址，因此最多可以表示 $2^{16} = 65536$ 个盘块。每个盘块大小为 4K，所以最大分区容量为 $2^{16} \times 4K = 65536 \times 4 \times 1024 \text{ Bytes} = 268435456 \text{ Bytes} = 256M$ 。

6.2 填空题（答案要求准确）

第 1 题 答案：40 (1 分)

解析：为保证响应时间不超过 2s (2000ms)，对于 50 个用户，在理想的轮转情况下，每个用户在一个周期内都能获得一次 CPU 时间。因此，时间片最大应为 $2000 \frac{\text{ms}}{50} = 40 \text{ ms}$ 。

第 2 题 答案：130 (1 分)

解析：程序 A 优先。A：计算 1(30ms)，I/O(40ms)，计算 2(10ms) B：计算 1(60ms)，I/O(30ms)，计算 2(10ms) 过程：0-30ms：A 执行计算 1。30-70ms：A 执行 I/O。此时 B 可以执行计算 1（从 30ms 开始）。B 的计算 1 需要 60ms，到 90ms 结束。70-80ms：A 执行计算 2（A 完成于 80ms）。90-120ms：B 执行 I/O（B 的计算 1 在 90ms 完成）。120-130ms：B 执行计算 2（B 完成于 130ms）。系统总运行时间为最后一个程序完成的时间，即 130ms。

第 3 题 答案：混合索引分配方式 (或 多级索引分配) (1 分)

解析：UNIX 系统采用的是混合索引分配方式，结合了直接地址、一级间接地址、二级间接地址和三级间接地址来管理文件块。

第 4 题 答案：2 (1 分)

解析：信号量 S.value 的含义：若 $S.value \geq 0$ ，表示可用资源的数量；若 $S.value < 0$ ，其绝对值表示等待队列中等待该资源的进程个数。 $S.value = -2$ ，表示有 2 个进程在等待打印机。

第 5 题 答案：消息缓冲区 (或 共享存储区/信箱) (1 分)

解析：消息缓冲队列通信机制属于间接通信方式，进程通过读写内核中提供的一个公用的消息缓冲区（也称信箱或共享内存区域中的特定结构）来进行通信。

第 6 题 答案：10153 (1 分)

解析：页面大小为 1K (1024 字节)。逻辑地址 2985。页号 $P = \lfloor \frac{2985}{1024} \rfloor = \lfloor 2.915 \rfloor = 2$ 。页内偏移量 $W = 2985 \bmod 1024 = 2985 - 2 \times 1024 = 2985 - 2048 = 937$ 。题目给出的映射关系是：第 0 页->块 2，第 1 页->块 6，第 2 页->块 9，第 3 页->块 7。逻辑页 2 对应物理块 9。物理地址 = 块号 \times 页面大小 + 页内偏移量 = $9 \times 1024 + 937 = 9216 + 937 = 10153$ 。

第 7 题 答案：DMA (直接存储器存取) (1 分)

解析：硬盘等高速 I/O 设备通常采用 DMA (Direct Memory Access) 方式进行数据传输，以减少 CPU 的干预，提高系统效率。

第 8 题 答案：19 (1 分)

解析：盘块大小 = 1KB。硬盘大小 = 500MB。总盘块数 = $500M \frac{B}{1KB} = \frac{500 \times 1024KB}{1} KB = 512000$ 个盘块。FAT 表项需要能够表示所有盘块的地址。设需要 n 个二进制位，则 $2^n \geq 512000$ 。 $2^{18} = 262144$ 。 $2^{19} = 524288$ 。因此，至少需要 19 个二进制位。

第 9 题 答案：127 (1 分)

解析：位示图共 30 行、16 列。行号、列号和盘块号均从 1 开始。第 8 行、第 15 列的盘块号计算公式为：盘块号 = (行号 - 1) \times 每行列数 + 列号。盘块号 = $(8 - 1) \times 16 + 15 = 7 \times 16 + 15 = 112 + 15 = 127$ 。

第 10 题 答案：互斥条件 (1 分)

解析：死锁产生的四个必要条件是：互斥条件、请求和保持条件、不可剥夺条件、循环等待条件。其中，互斥条件是由许多资源的固有特性（如打印机一次只能一个进程使用）决定的，一般不能破坏它，否则无法保证数据的一致性和正确性。

6.3 简答题 (共 3 题, 每题 5 分, 共 15 分)

第 1 题答案 (5 分)

进程与程序的主要区别如下：

1. **定义：** 程序是静态的指令集合，是一组有序的指令代码，存储在磁盘等介质上。进程是程序的一次动态执行过程，是操作系统进行资源分配和调度的基本单位。
2. **状态：** 程序是静态的，没有状态。进程是动态的，具有多种状态（如创建、就绪、运行、阻塞、终止）。
3. **生命周期：** 程序可以永久存在。进程有生命周期，会创建、执行和消亡。
4. **组成：** 程序主要指代码。进程由程序段、数据段和进程控制块（PCB）组成。PCB 是进程存在的唯一标志。
5. **对应关系：** 一个程序可以对应多个进程（例如运行一个程序多次），一个进程至少包含一个程序段。

第 2 题答案 (5 分)

什么是虚拟存储器： 虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量进行扩充的一种存储器系统。其逻辑容量由 CPU 的寻址范围决定，其运行速度接近于主存速度，而成本又接近于辅存成本。它使得用户编程不受实际物理内存大小的限制，程序可以比物理内存大。

基本特征：

1. **多次性：** 一个作业被分成多次调入内存运行，无需一次全部装入。
2. **对换性（或置换性）：** 作业运行时无需一直常驻内存，允许将暂时不用的程序或数据部分调出到外存，需要时再调入。
3. **虚拟性：** 能够从逻辑上扩充内存容量，用户看到的内存容量远大于实际物理内存容量。
4. **离散性：** 程序和数据可以离散地存放在内存的非连续区域中。

第 3 题答案 (5 分)

什么是进程同步： 进程同步是指在多道程序环境下，为了协调多个并发进程的执行顺序，确保它们能有效地共享资源和相互协作，从而避免出现数据不一致或运行结果错误等问题，而对进程的执行次序所进行的一种协调。

同步机制应遵循的原则：

1. **空闲让进：** 当没有进程处于临界区时，任何一个请求进入临界区的进程都应能立即进入。
2. **忙则等待：** 当已有进程进入其临界区时，其他试图进入临界区的进程必须等待。
3. **有限等待：** 对要求访问临界资源的进程，应保证其在有限时间内能进入临界区，避免“饿死”。
4. **让权等待（可选，但提倡）：** 当进程不能进入其临界区时，应立即释放处理机，以免进程陷入“忙等”。

6.4 算法综合题 (共 6 题, 共 50 分)

第 1 题解答过程 (共 10 分)

解： 进程信息：A(0,3), B(2,5), C(3,4), D(5,4), E(8,2) 周转时间 = 完成时间 - 到达时间 带权周转时间 = 周转时间 / 服务时间

1. **先来先服务 (FCFS) 调度算法：** 调度顺序：A → B → C → D → E

- 进程 A: 到达 0, 服务 3. 完成时间 = 3. 周转时间 = 3 - 0 = 3. 带权周转时间 = $\frac{3}{3} = 1.0$.
- 进程 B: 到达 2, 服务 5. 开始时间 = 3. 完成时间 = 3 + 5 = 8. 周转时间 = 8 - 2 = 6. 带权周转时间 = $\frac{6}{5} = 1.2$.
- 进程 C: 到达 3, 服务 4. 开始时间 = 8. 完成时间 = 8 + 4 = 12. 周转时间 = 12 - 3 = 9. 带权周转时间 = $\frac{9}{4} = 2.25$.
- 进程 D: 到达 5, 服务 4. 开始时间 = 12. 完成时间 = 12 + 4 = 16. 周转时间 = 16 - 5 = 11. 带权周转时间 = $\frac{11}{4} = 2.75$.
- 进程 E: 到达 8, 服务 2. 开始时间 = 16. 完成时间 = 16 + 2 = 18. 周转时间 = 18 - 8 = 10. 带权周转时间 = $\frac{10}{2} = 5.0$.

2. **高响应比优先 (HRRN) 调度算法：** 响应比 = (等待时间 + 服务时间) / 服务时间

- 时间 0: 只有 A 到达，A 执行。A: 完成时间 = 3. 周转时间 = 3 - 0 = 3. 带权周转时间 = $\frac{3}{3} = 1.0$.
- 时间 3: A 完成。B(到达 2, 等待 1), C(到达 3, 等待 0)已到达。RR(B) = $\frac{1+5}{5} = 1.2$. RR(C) = $\frac{0+4}{4} = 1.0$. 选择 B 执行。B: 开始 3, 完成 3 + 5 = 8. 周转时间 = 8 - 2 = 6. 带权周转时间 = $\frac{6}{5} = 1.2$.
- 时间 8: B 完成。C(到达 3, 等待 5), D(到达 5, 等待 3), E(到达 8, 等待 0)已到达。RR(C) = $\frac{5+4}{4} = 2.25$. RR(D) = $\frac{3+4}{4} = 1.75$. RR(E) = $\frac{0+2}{2} = 1.0$. 选择 C 执行。C: 开始 8, 完成 8 + 4 = 12. 周转时间 = 12 - 3 = 9. 带权周转时间 = $\frac{9}{4} = 2.25$.
- 时间 12: C 完成。D(到达 5, 等待 7), E(到达 8, 等待 4)已到达。RR(D) = $\frac{7+4}{4} = 2.75$. RR(E) = $\frac{4+2}{2} = 3.0$. 选择 E 执行。E: 开始 12, 完成 12 + 2 = 14. 周转时间 = 14 - 8 = 6. 带权周转时间 = $\frac{6}{2} = 3.0$.
- 时间 14: E 完成。只剩 D(到达 5, 等待 9)。选择 D 执行。D: 开始 14, 完成 14 + 4 = 18. 周转时间 = 18 - 5 = 13. 带权周转时间 = $\frac{13}{4} = 3.25$.

调度算法	进程 A	进程 B	进程 C	进程 D	进程 E
FCFS					
完成时间	3.0	8.0	12.0	16.0	18.0
周转时间	3.0	6.0	9.0	11.0	10.0
带权周转时间	1.0	1.2	2.25	2.75	5.0
HRRN					
完成时间	3.0	8.0	12.0	18.0	14.0
周转时间	3.0	6.0	9.0	13.0	6.0
带权周转时间	1.0	1.2	2.25	3.25	3.0

表 6 表 3 调度结果 (以小数方式表示)

答案: 结果见上表

第 2 题解答过程 (共 10 分 (5+3+2))

初始状态: Available = (1,6,2,2) Allocation: P0(0,0,3,2), P1(1,0,0,0), P2(1,3,5,4), P3(0,3,3,2), P4(0,0,1,4) Need: P0(0,0,1,2), P1(1,7,5,0), P2(2,3,5,6), P3(0,6,5,2), P4(0,6,5,6)

(1) 该状态是否安全? 解: 使用安全性算法检查。Work = Available = (1,6,2,2); Finish = [F,F,F,F,F]

- 步骤 1: P0. Need P0(0,0,1,2) ≤ Work(1,6,2,2). True. Work = (1,6,2,2) + (0,0,3,2) = (1,6,5,4). Finish[P0]=T. 安全序列: <P0>
- 步骤 2: P1. Need P1(1,7,5,0) ≤ Work(1,6,5,4). False (7>6).
- 步骤 3: P2. Need P2(2,3,5,6) ≤ Work(1,6,5,4). False (2>1 or 6>4).
- 步骤 4: P3. Need P3(0,6,5,2) ≤ Work(1,6,5,4). True. Work = (1,6,5,4) + (0,3,3,2) = (1,9,8,6). Finish[P3]=T. 安全序列: <P0,P3>
- 步骤 5: P4. Need P4(0,6,5,6) ≤ Work(1,9,8,6). True. Work = (1,9,8,6) + (0,0,1,4) = (1,9,9,10). Finish[P4]=T. 安全序列: <P0,P3,P4>
- 步骤 6: P1. Need P1(1,7,5,0) ≤ Work(1,9,9,10). True. Work = (1,9,9,10) + (1,0,0,0) = (2,9,9,10). Finish[P1]=T. 安全序列: <P0,P3,P4,P1>
- 步骤 7: P2. Need P2(2,3,5,6) ≤ Work(2,9,9,10). True. Work = (2,9,9,10) + (1,3,5,4) = (3,12,14,14). Finish[P2]=T. 安全序列: <P0,P3,P4,P1,P2>

所有进程均可完成, 该状态安全。安全序列例如 P0, P3, P4, P1, P2。

(2) 若进程 P2 提出请求 Request(1,2,2,2), 系统能否将资源分配给它? 解:

- 检查 Request_P2(1,2,2,2) ≤ Need_P2(2,3,5,6). True.
- 检查 Request_P2(1,2,2,2) ≤ Available(1,6,2,2). True.
- 假设分配: New_Available = (1,6,2,2) - (1,2,2,2) = (0,4,0,0). New_Allocation_P2 = (1,3,5,4) + (1,2,2,2) = (2,5,7,6). New_Need_P2 = (2,3,5,6) - (1,2,2,2) = (1,1,3,4).
- 对新状态进行安全性检查: Work = (0,4,0,0). Need' P2(1,1,3,4). P0: Need(0,0,1,2) ≤ Work(0,4,0,0) → False (1>0). P1: Need(1,7,5,0) ≤ Work(0,4,0,0) → False (1>0). P2': Need(1,1,3,4) ≤ Work(0,4,0,0) → False (1>0). P3: Need(0,6,5,2) ≤ Work(0,4,0,0) → False (6>4). P4: Need(0,6,5,6) ≤ Work(0,4,0,0) → False (6>4). 新状态不安全。系统不能分配。

(3) 如果系统立即满足 P2 的请求, 系统是否会立即进入死锁状态? 为什么? 解: 不会立即进入死锁状态。不安全状态意味着存在进入死锁的风险, 但并非立即死锁。死锁是进程循环等待资源且无法继续执行的状态。银行家算法的目的是避免进入不安全状态。

答案: (1) 该状态安全, 安全序列例如 P0, P3, P4, P1, P2。(2) 系统不能分配, 因为分配后系统将进入不安全状态。(3) 不会立即进入死锁状态, 但系统进入不安全状态, 有发生死锁的风险。

第 3 题解答过程 (共 10 分)

页面走向: 3,4,1,3,2,0,3,2,4,2,3,0,1,2 (14 次)。物理块 M=4。

1. FIFO 页面置换算法:

访问序列	内存状态 (队头→队尾)	是否缺页	换出页
3	3	Y	-
4	3, 4	Y	-
1	3, 4, 1	Y	-
3	3, 4, 1	N	-
2	3, 4, 1, 2	Y	-

0	4, 1, 2, 0	Y	3
3	1, 2, 0, 3	Y	4
2	1, 2, 0, 3	N	—
4	2, 0, 3, 4	Y	1
2	2, 0, 3, 4	N	—
3	2, 0, 3, 4	N	—
0	2, 0, 3, 4	N	—
1	0, 3, 4, 1	Y	2
2	3, 4, 1, 2	Y	0

FIFO 缺页次数: 9 次。缺页率: $\frac{9}{14} \approx 0.64$ 。

2. LRU 页面置换算法: (栈底为最久未使用, 栈顶为最近使用)

访问序列	内存状态 (栈底→栈顶)	是否缺页	换出页
3	3	Y	—
4	3, 4	Y	—
1	3, 4, 1	Y	—
3	4, 1, 3	N	—
2	4, 1, 3, 2	Y	—
0	1, 3, 2, 0	Y	4
3	1, 2, 0, 3	N	—
2	1, 0, 3, 2	N	—
4	0, 3, 2, 4	Y	1
2	0, 3, 4, 2	N	—
3	0, 4, 2, 3	N	—
0	4, 2, 3, 0	N	—
1	2, 3, 0, 1	Y	4
2	3, 0, 1, 2	N	—

LRU 缺页次数: 7 次。缺页率: $\frac{7}{14} = 0.50$ 。

答案: FIFO 算法: 缺页次数: 9 次。缺页率: $\frac{9}{14} \approx 0.64$ 。LRU 算法: 缺页次数: 7 次。缺页率: $\frac{7}{14} = 0.50$ 。

第 4 题解答过程 (共 10 分)

磁道 0-199。当前 55。请求队列: 100, 183, 40, 122, 13, 124, 65, 69, 90, 50。SCAN 先向磁道号减少方向。

1. FCFS 算法: 服务次序: 55 → 100 → 183 → 40 → 122 → 13 → 124 → 65 → 69 → 90 → 50 寻道长度: $|100 - 55| + |183 - 100| + |40 - 183| + |122 - 40| + |13 - 122| + |124 - 13| + |65 - 124| + |69 - 65| + |90 - 69| + |50 - 90| = 45 + 83 + 143 + 82 + 109 + 111 + 59 + 4 + 21 + 40 = 697$ 道。平均寻道长度 = $\frac{697}{10} = 69.7$ 道。

2. SCAN 算法 (先向磁道号减少方向): 服务次序: 55 → 50 → 40 → 13 → 0 (到达边界) → 65 → 69 → 90 → 100 → 122 → 124 → 183 寻道长度: $(55 - 50) + (50 - 40) + (40 - 13) + (13 - 0) + (65 - 0) + (69 - 65) + (90 - 69) + (100 - 90) + (122 - 100) + (124 - 122) + (183 - 124) = 5 + 10 + 27 + 13 + 65 + 4 + 21 + 10 + 22 + 2 + 59 = 238$ 道。(Alternatively: $(55 - 0) + (183 - 0) = 55 + 183 = 238$ 道) 平均寻道长度 = $\frac{238}{10} = 23.8$ 道。

答案: FCFS 算法: 服务次序: 55 → 100 → 183 → 40 → 122 → 13 → 124 → 65 → 69 → 90 → 50。总寻道长度: 697 道。平均寻道长度: 69.7 道。SCAN 算法: 服务次序: 55 → 50 → 40 → 13 → 0 → 65 → 69 → 90 → 100 → 122 → 124 → 183。总寻道长度: 238 道。平均寻道长度: 23.8 道。

第 5 题答案 (5 分)

根据文件路径和图示的目录结构:

- A 是 home (因为路径如 /home/wang/...)

- B 是 sun (因为路径如 /home/sun/... 且 A 下有 wang, zhang, B)
- C 是 test (因为路径 /home/wang/test/work 且 wang 下是 C)
- D 是 mydir (因为路径 /home/sun/mydir/work 且 sun 下是 D)
- E 是共享文件 work 的 i-node。

所以： A: home B: sun C: test D: mydir E: work (文件 i-node)

第 6 题答案 (5 分)

文件 A 依次占用磁盘块号：3, 6, 8, 10, 5。MS-DOS 文件系统使用文件分配表 (FAT) 来链接文件的各个盘块。

文件 A 的盘块链接情况：起始块: 3 链接: 3 → 6 → 8 → 10 → 5 (结束)

FAT 表示意 (假设文件结束符为 -1 或 FFF 等):

FAT 表地址 | 内容 (下一块号 / 结束标志)

...	...
2	(其他文件/空闲)
3	6
4	(其他文件/空闲)
5	-1 (文件结束)
6	8
7	(其他文件/空闲)
8	10
9	(其他文件/空闲)
10	5
...	...

图示：目录条目中文件 A 的起始块号指向 3。(逻辑磁盘块) [块 3] → [块 6] → [块 8] → [块 10] → [块 5 (EOF)]

FAT 表中，索引为 3 的条目存放 6，索引为 6 的条目存放 8，索引为 8 的条目存放 10，索引为 10 的条目存放 5，索引为 5 的条目存放文件结束标志。

6.5 程序设计题 (共 1 题, 每题 10 分, 共 10 分)

第 1 题参考代码 (共 10 分)



```
// 定义信号量和缓冲区 (伪代码风格)
semaphore mutex1 = 1; // 缓冲池1互斥锁
semaphore empty1 = m; // 缓冲池1空缓冲区数
semaphore full1 = 0; // 缓冲池1满缓冲区数
buffer pool1[m]; // P和Q共享的缓冲池1

semaphore mutex2 = 1; // 缓冲池2互斥锁
semaphore empty2 = n; // 缓冲池2空缓冲区数
semaphore full2 = 0; // 缓冲池2满缓冲区数
buffer pool2[n]; // Q和R共享的缓冲池2

// 假设 item_type 为传输的信息单元类型
item_type data_item;

// 进程P: 从输入设备读信息, 放入pool1
process P {
    while (true) {
        read_from_input_device(&data_item); // 从输入设备读取信息

        P(empty1); // 等待pool1有空闲缓冲区
        P(mutex1); // 进入临界区
        add_item_to_buffer(pool1, data_item); // 信息放入pool1
        V(mutex1); // 退出临界区
        V(full1); // 通知pool1有数据了
    }
}

// 进程Q: 从pool1取信息, 加工后放入pool2
process Q {
    item_type item_from_p, item_for_r;
    while (true) {
        P(full1); // 等待pool1有数据
```

```

P(mutex1);    // 进入临界区
remove_item_from_buffer(pool1, &item_from_p); // 从pool1取出信息
V(mutex1);    // 退出临界区
V(empty1);    // 通知pool1有空位了

process_data(item_from_p, &item_for_r); // 加工信息

P(empty2);    // 等待pool2有空闲缓冲区
P(mutex2);    // 进入临界区
add_item_to_buffer(pool2, item_for_r); // 加工后信息放入pool2
V(mutex2);    // 退出临界区
V(full2);     // 通知pool2有数据了
}
}

// 进程R: 从pool2取信息并打印输出
process R {
    item_type item_to_print;
    while (true) {
        P(full2);    // 等待pool2有数据
        P(mutex2);    // 进入临界区
        remove_item_from_buffer(pool2, &item_to_print); // 从pool2取出信息
        V(mutex2);    // 退出临界区
        V(empty2);    // 通知pool2有空位了

        print_output(item_to_print); // 打印输出信息
    }
}

```

评分要点:

- 正确定义和初始化所有必需的信号量 (mutex1, empty1, full1, mutex2, empty2, full2)。(3 分)
- 进程 P 的逻辑正确: P(empty1) → P(mutex1) → 操作 → V(mutex1) → V(full1)。(2 分)
- 进程 Q 的逻辑正确:
 - 从 pool1 取: P(full1) → P(mutex1) → 操作 → V(mutex1) → V(empty1)。
 - 向 pool2 放: P(empty2) → P(mutex2) → 操作 → V(mutex2) → V(full2)。(3 分)
- 进程 R 的逻辑正确: P(full2) → P(mutex2) → 操作 → V(mutex2) → V(empty2)。(2 分)
- PV 操作成对出现, 同步信号量在外, 互斥信号量在内 (保护临界区)。