

《操作系统》复习纲要

考试内容涉及课本第一章至第十章所有讲授内容；

考试说明：

1. 闭卷。
2. 不允许使用计算器等电子设备。
3. 无需带草稿纸。
4. 答案写在答卷的对应栏中。是答卷不是白纸！

考试

- 题目类型
 - 选择题(10%)
 - 是非判断题(10%)
 - 简答题(20%)
 - 综合题(60%)

第一章 概述

- 操作系统的定义
- 提供用户的使用方式
- 五大功能；
- 操作系统的分类、各种类别的特点；
- 特权指令、处理器的状态及程序状态字。
- 操作系统结构
- 现代操作系统特点

第二章 进程与线程

- 多道程序并发执行的特点；
- 进程的定义及其与程序的区别、进程的组成、进程的三态及转换、进程控制原语
- 线程的定义、与进程的区别。
- 多核、多线程与超线程

第三章 互斥与同步

- 并发进程产生与时间有关错误;
- 临界区, 对临界区的管理要求;
- 信号量的定义, 其物理意义;
- 原语概念;
- 用信号量实现进程互斥与同步;
- 直接通信, 间接通信;
- 死锁的定义, 四个必要条件;
- 死锁的防止;
- 死锁的避免与银行家算法;
- 死锁的检测与恢复。

信号量设置及初值设置

- 公有信号量：互斥—间接制约关系
 - 如多个进程共享打印机
 - semaphore mutex=1
- 私有信号量：同步—直接制约关系
 - 如生产者与消费者
 - semaphore sp=K, sc=0
 - 注意：
 - sc=0：意味着消费者需要等待生产者放完产品后方可取产品，即只有生产者放产品后消费者才可取产品

互斥与同步例题

有5个哲学家，他们的生活方式是交替地思考和进餐。哲学家们共用一张圆桌，围绕着圆桌而坐，在圆桌上有6个碗和5支筷子，平时哲学家进行思考，饥饿时拿起其左、右的两支筷子，试图进餐，进餐完毕又进行思考，如图所示。这里的问题是哲学家只有拿到靠近他的两支筷子才能进餐，而拿到两支筷子的条件是他的左、右邻居此时都没有进餐。写出不存在死锁的同步与互斥算法



```
semaphore chopstick[5]={1, 1, 1, 1, 1};  
void philosopher(int i)    /*哲学家进程*/  
{  
    while (true) {    资源层次分配方法。  
        if (i<i+1)%5){ P(chopstick[i]); P(chopstick[(i+1)%5]); }  
        else{ P(chopstick[(i+1)%5]); P(chopstick[i]); }  
        ...;  
        eat;    /*进餐*/  
        ...:  
        V(chopstick[i]);    V(chopstick[(i+1)%5]);  
        ...:  
        think;    /*思考*/  
        ...;  
    }  
}
```



```
semaphore chopstick[5]={1, 1, 1, 1, 1}, s=4;
void philosopher(int i)    / *哲学家进程* /
{ while (true) {
    P(s);                  死锁避免; 银行家算法?
    P(chopstick[i]); P(chopstick[(i+1)%5]);
    ...;
    eat;    / *进餐* /
    ...:
    V(chopstick[i]);    V(chopstick[(i+1)%5]);
    V(s);
    ...:
    think;    / *思考* /
    ...;
}
}
```

死锁例题

某一系统在T0时刻进程的资源分配“瞬间状态”为：

进程	已占有资源量				最大资源需求量				剩余资源量			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P1	0	0	0	1	5	5	3	2	3	2	1	0
P2	1	1	2	3	3	4	5	5				
P3	2	1	1	0	4	3	5	6				
P4	1	2	3	2	4	4	4	2				

使用银行家算法回答：

(1)系统是否安全？如果安全，请给出一个安全执行序列，如果不安全，说明理由。要求给出计算过程。

(2)如果在T0时刻进程P1申请资源(0, 1, 1, 1)，系统能否立即满足进程的要求？请说明理由。

【解】(1)、安全的。(2分)

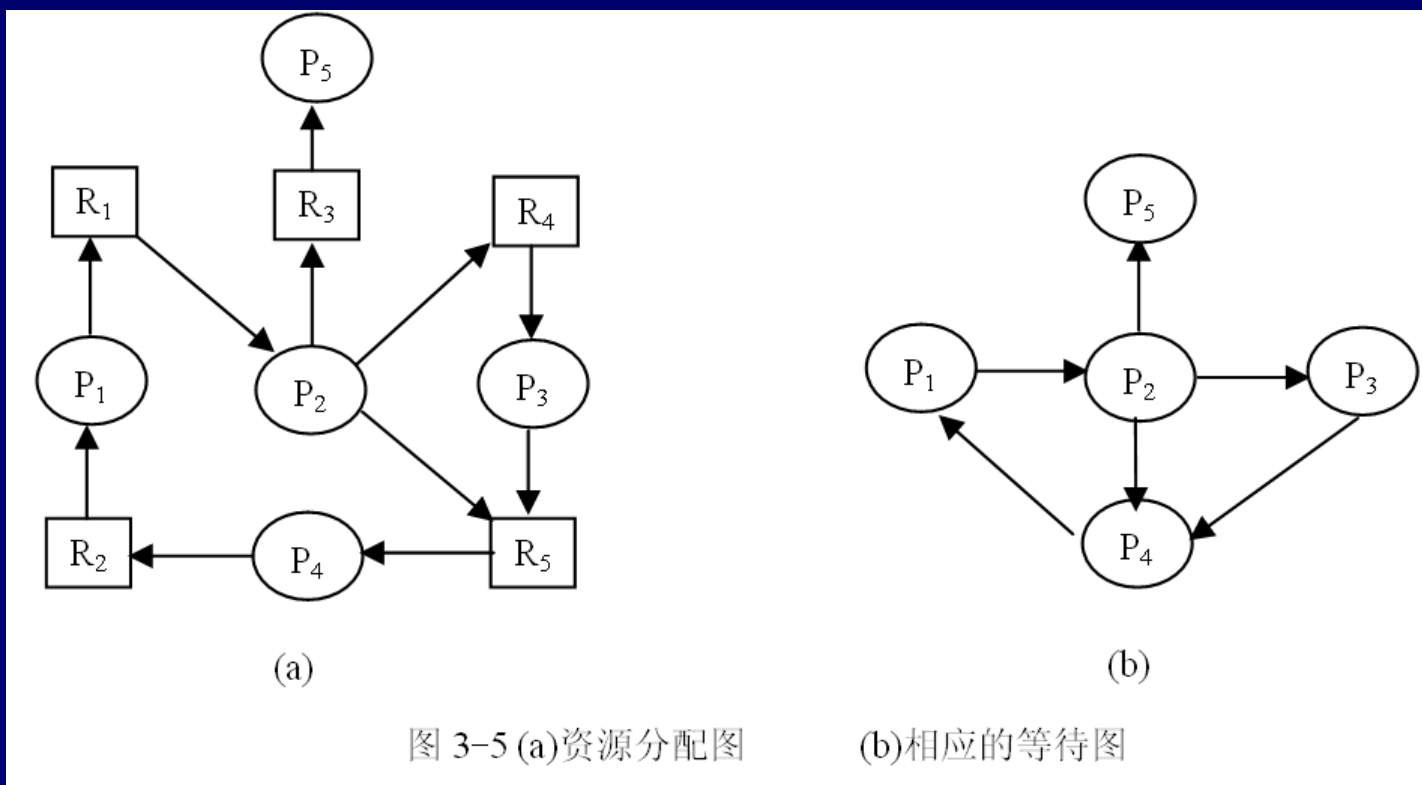
先计算各个进程还需要的资源：

P1: (5 5 3 1), P2:(2 3 3 2), P3: (2 2 4 6), P4:(3 2 1 0);

系统剩余资源为(3 2 1 0)，仅能满足进程P4对资源的最大需要，故分配P4所需要的资源，P4在得到所需全部资源后应该在有限的时间内执行结束，并归还它所占用的全部资源；系统现在可用资源为(4 4 4 2)，仅能满足进程P2对资源的最大需要，在P2执行结束并归还它所占用的全部资源后系统资源为(5 5 6 5)，此时可满足P1对资源的最大需求，在P1执行结束并归还它所占用的全部资源后系统资源为(5 5 6 6)，此时可满足P3对资源的最大需求，因此系统是安全的，安全执行序列为P4P2P1P3(6分)。

(2)、若在T0时刻进程P1申请(0, 1, 1, 1)，系统不能立即满足进程的要求。因为，虽然系统剩余资源能够满足进程P1当前对资源申请需求，但不满足其对资源的最大需求，如果分配给P1，那么系统剩余资源不能满足任何一个进程对资源的最大需求，没有一个进程能够安全执行结束，从而造成相互等待的情况，导致系统的不安全性(4分)。

死锁检测：资源分配图和进程等待图



第四章 处理器调度

- 处理器调度层次
- 调度准则
- 短程调度算法
 - 单处理器调度算法
 - 多处理器调度算法
- 实时调度

第五章 存储管理

- 存储管理的功能、存储分配的三种方式；
- 重定位的定义、两种重定位的特点与区别、覆盖与交换；
- 分区存储管理、页式存储管理
(各种方法采用的分配回收算法，数据结构，地址变换过程，共享与保护，优缺点比较)
- 虚拟存储器基本思想，页式虚拟存储工作流程，
- 常用的页面置换算法（FIFO, LRU, OPT, CLOCK, NRU, LFU）
- 段式存储管理的思想，段式虚拟存储管理流程。

访问次序	1	2	3	4	5	1	6	2	1	5	6	2	5	2	3	5
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
淘汰							3				4				1	

【解】FIFO 共10次缺页中断，缺页中断率=10/16=62.5%

例：设某计算机的逻辑地址空间和物理地址空间均为64KB，按字节编址。若某进程最多需要6页数据存储空间，页的大小为4KB。操作系统采用固定分配局部置换策略为此进程分配4个页框。如果该进程依次要访问的地址序列是：0123H，1234H，2345H，3456H，1357H，0586H，4ABCH，3088H，1235H，0987H。试问分别采用FIFO、LRU和OPT算法时，各产生多少次缺页中断？并计算相应的缺页中断率，同时写出在这三种调度算法下产生缺页中断时淘汰的页面号和在主存的页面号。

解：根据页式管理的工作原理，应先考虑页面大小，以便将页号和页内位移分解出来。页面大小为4KB，即 2^{12} ，则得到页内位移占虚拟地址的低12位，页号占剩余高位。因此地址序列：0123H，1234H，2345H，3456H，1357H，0586H，4ABCH，3088H，1235H，0987H对应的页号是0,1,2,3,1,0,4,3,1,0。（3分）

【解】FIFO 共6次缺页中断
，缺页中断率
=6/10=60%（3分）

访问次序	0	1	2	3	1	0	4	3	1	0
主存页号				3	3	3	3	3	3	3
			2	2	2	2	2	2	2	2
		1	1	1	1	1	1	1	1	0
	0	0	0	0	0	0	4	4	4	4
淘汰							0			1

LRU 共5次缺页中断
，缺页中断率
=5/10=50%（3分）

访问次序	0	1	2	3	1	0	4	3	1	0
主存页号				3	1	0	4	3	1	0
			2	2	3	1	0	4	3	1
		1	1	1	2	3	1	0	4	3
	0	0	0	0	0	2	3	1	0	4
淘汰							2			

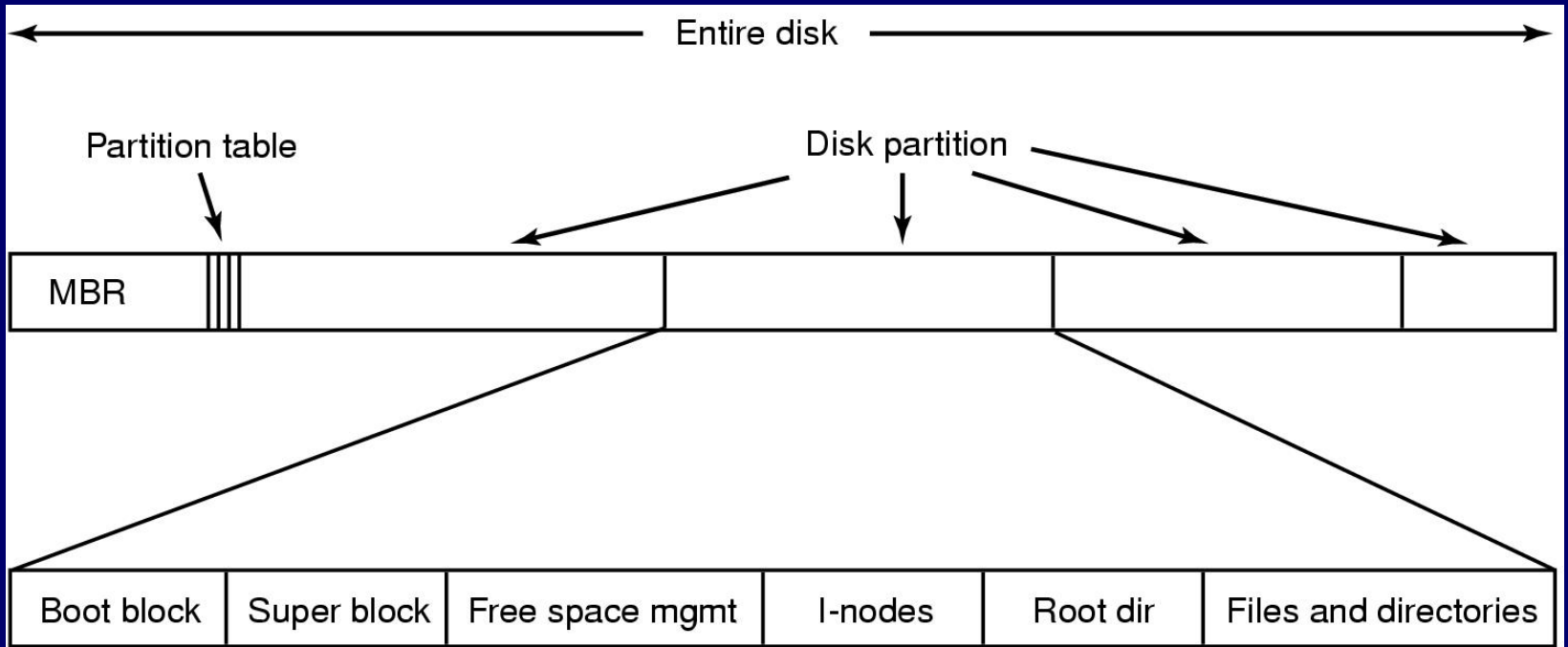
OPT 共5次缺页中断
，缺页中断率
=5/10=50%（3分）

访问次序	0	1	2	3	1	0	4	3	1	0
主存页号				3	3	3	3	3	3	3
			2	2	2	2	4	4	4	4
		1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0
淘汰							2			

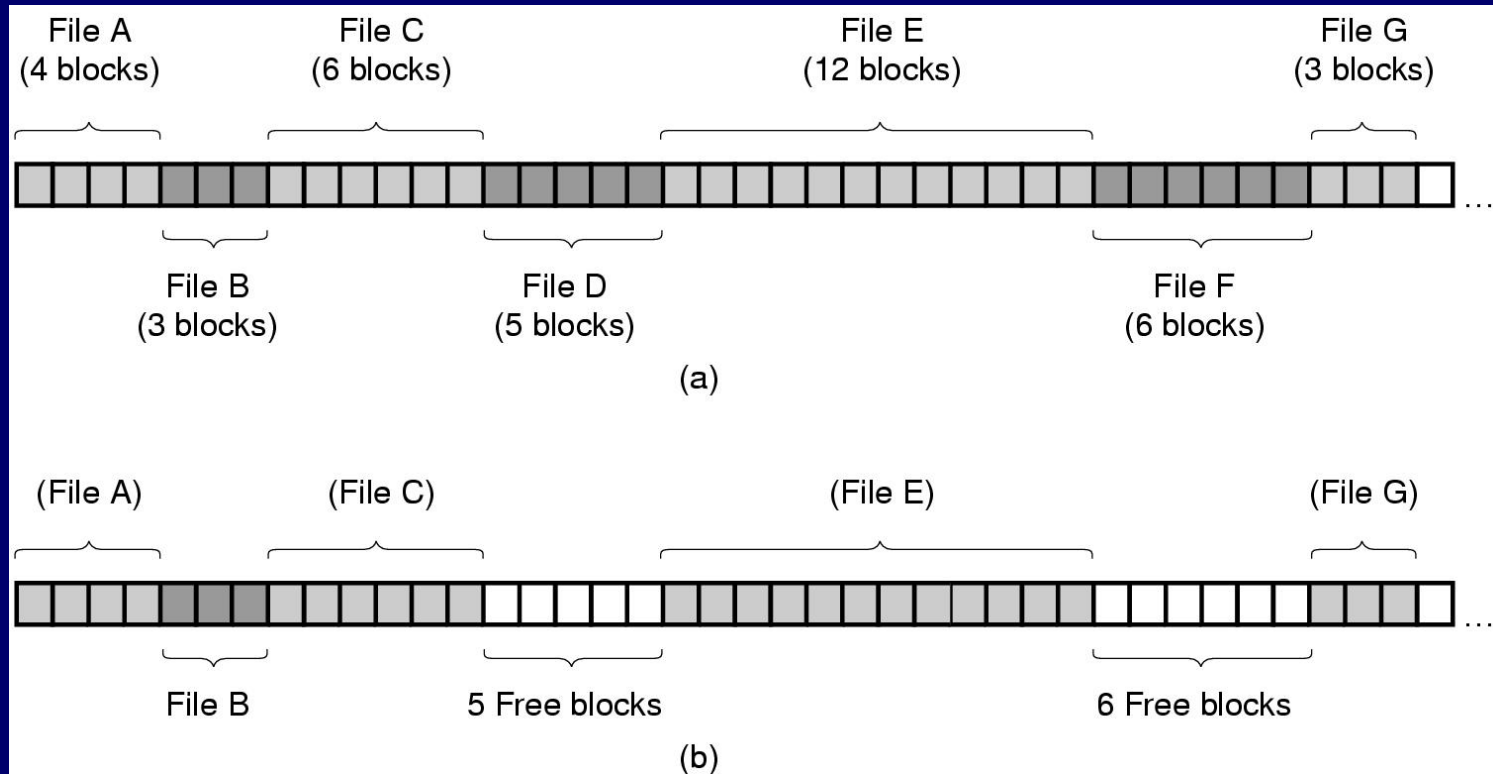
第六章 文件管理

- 文件
- 文件目录
- 文件共享与保护
- 文件存储空间的管理
- 文件分配

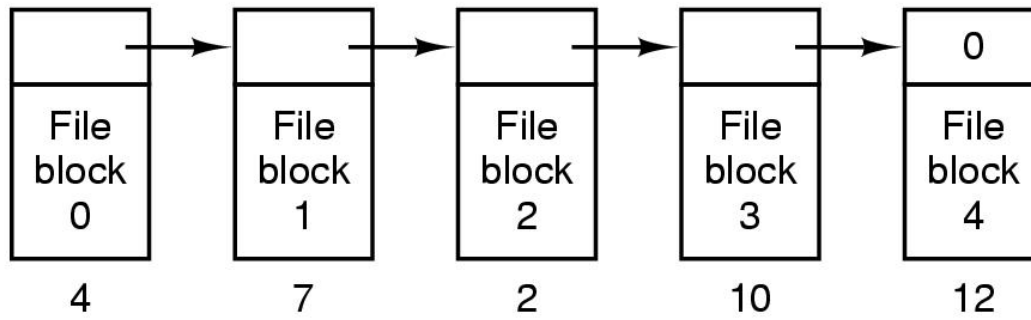
第六章 文件管理



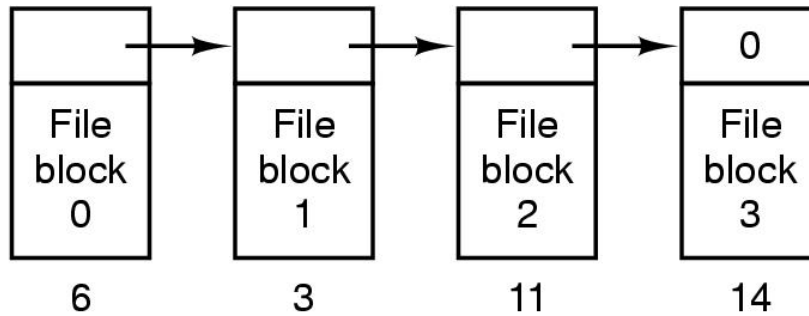
第六章 文件管理



File A

Physical
block

File B

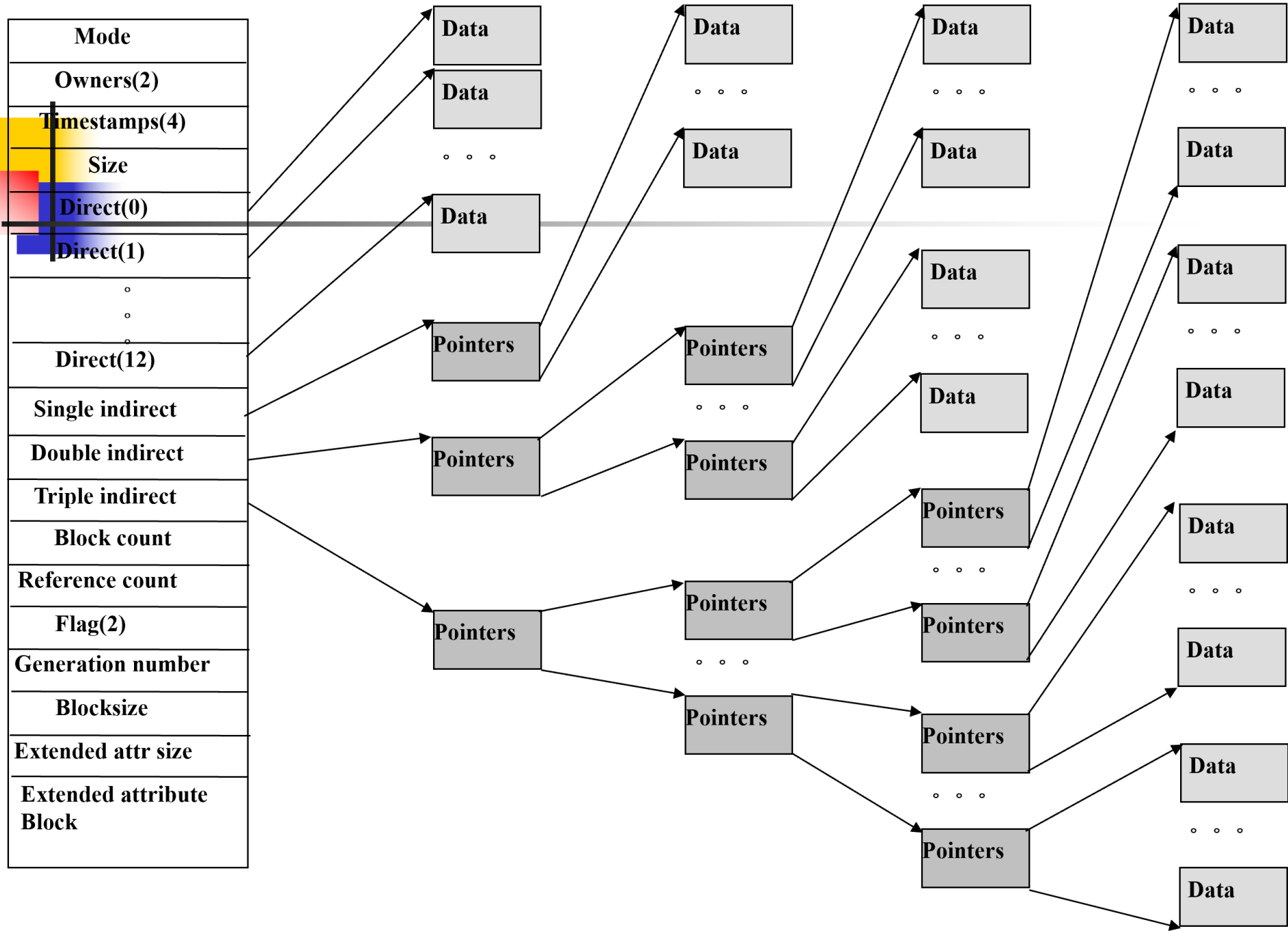
Physical
blockPhysical
block

0	
1	
2	10
3	11
4	7
5	
6	3
7	2
8	
9	
10	12
11	14
12	-1
13	
14	-1
15	

← File A starts here

← File B starts here

← Unused block



A UNIX file system has 4-KB blocks and 4-byte disk addresses. Assume that there are 10 direct block pointers, and a singly, doubly, and triply indirect pointer in each i-node, then

- a. What is the maximum file size supported by this system?**
- b. Assuming no information other than that the file i-node is already in main memory, how many disk accesses are required to access the byte in position 12345?**
- c. Assuming no information other than that the file i-node is already in main memory, how many disk accesses are required to access the byte in position 1234567890?**

(1) $4K/4=1K$

$10*4K+1K*4K+1K*1K*4K+1K*1K*1K*4K=4T4G4M40KB$

(2) 因为 $12345 < 40K$ ，故该地址落入直接索引中，由于i-node已在内存，因此只需要1次访问磁盘块，即一次访问含有该位置数据的数据块。(3分)

(3) 因为 $4M40K < 1234567890 < 4G4M40K$ ，故该地址落入二级间接索引中，由于i-node已在内存，因此只需要3次访问磁盘块。

第七章 I/O管理

- I/O管理概述
- I/O控制方式
- I/O缓冲
- 设备驱动程序
- 设备分配
- 磁盘存储器的管理,常用调度算法
(FCFS,SSTF,SCAN,C-SCAN);
- 磁盘阵列(RAID)
- 时钟管理和电源管理

设某磁盘旋转速度为每分钟6000转。每个磁道有100个扇区，相邻磁道间的平均移动时间为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服务，磁道号请求队列为50、200、80、110、180，对请求队列中的每个磁道需读取1个随机分布的扇区，则：

(1)在SSTF(最短寻道)调度情况下，读完这5个扇区总共需要多少时间，要求给出计算过程；

(2)在SCAN(扫描)调度情况下，读完这5个扇区总共需要多少时间，要求给出计算过程；

(3)就以上二种算法进行对比分析，并为用户选择提供建议说明。

【解】由于转速为6000r/m，则平均旋转延迟为 $60 \times 1000 / (6000 \times 2) = 5\text{ms}$ ，总的旋转延迟时间为 $5 \times 5\text{ms} = 25\text{ms}$ 。

由于旋转一周的时间为10ms，每个磁道有100个扇区，读取一个磁道上一个扇区的平均读取时间为 $10\text{ms} / 100 = 0.1\text{ms}$ ，总的读取扇区的时间 $= 0.1\text{ms} \times 5 = 0.5\text{ms}$ 。

1) SSTF:执行顺序是100->110->80->50->180->200。则移到磁道长度为 $10 + 60 + 150 = 220$ 。故读取上述磁道上所有扇区所花的总时间 $= 220\text{ms} + 25\text{ms} + 0.5\text{ms} = 245.5\text{ms}$

2) 采用SCAN调度算法访问磁道的顺序为100->110->180->200->80->50，则移到磁道长度为 $100 + 150 = 250$ 。

故读取上述磁道上所有扇区所花的总时间 $= 250\text{ms} + 25\text{ms} + 0.5\text{ms} = 275.5\text{ms}$

两者算法都有寻道时间短、寻道优化的优势，而且两种寻道时间相差30ms，但相比较而言，SSTF可能会频繁切换服务方向，而SCAN不会。由于磁臂调度是机械运动，存在惯性，频繁切换服务方向会降低机械装置的使用寿命，故实际应用中SCAN优于SSTF。

第八章 操作系统的安全性

- 安全的计算机系统的三个特性：
- 保密性、完整性和可用性
- OS的安全性是计算机系统安全的基础
- 有选择的访问控制、内存管理与对象重用、审计能力、文件和数据的加密以及进程间的安全通信机制。
- 保护机制：保护域、ACL、权能表
- 安全机制：
- 用户认证；恶意代码及时防范、发现和清除；使用情况的监控和审计；加密技术；备份。

第九章 新型操作系统

- 嵌入式操作系统
 - 微软嵌入式操作系统
 - 嵌入式Linux与安卓
 - TinyOS
 - VxWorks
- 多媒体操作系统
- 分布式操作系统
 - 集群系统
 - MapReduce计算模型
- 虚拟化技术

第十章 操作系统设计问题

- 操作系统设计目标
 - 抽象定义
 - 原语操作提供
 - 隔离性保证
 - 硬件管理
- 界面设计
 - 界面设计的原则
 - 界面设计的范式
 - 系统调用界面
- 操作系统设计实现
- 性能优化
- 项目管理