第一章操作系统概论

1-5 BCBDA　　6-10 DDCBC

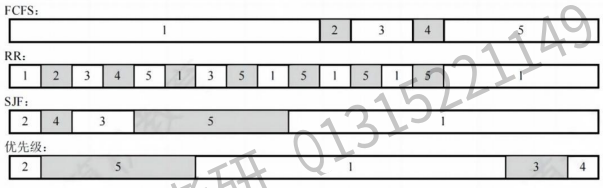
第二章进程与线程

1-5 CACDC　　6-10 BBCAC　　11-15ADBDA　　16-20 CDDBB

21-25 CDBDB　　26-28 BBD　　32-36 DDACD　　37-41 CAD(CB)B

48-49 CB　　51-55 DBBBB　　56-60 DCDCC　　61-62DB

29. 1）作业执行情况可以用如下的甘特图来表示。



2）各个作业对应于各个算法的周转时间和加权周转时间见下表。

算法	时间类型	P_1	P_2	P_3	P_4	P_5	平均时间
	运行时间	10	1	2	1	5	3.8
FCFS	周转时间	10	11	13	14	19	13.4
	加权周转时间	1	11	6.5	14	3.8	7.26
RR	周转时间	19	2	7	4	14	9.2
	加权周转时间	1.9	2	3.5	4	2.8	2.84
SJF	周转时间	19	1	4	2	9	7
	加权周转时间	1.9	1	2	2	1.8	1.74
优先级	周转时间	16	1	18	19	6	12
	加权周转时间	1.6	1	9	19	1.2	6.36

所以，FCFS 的平均周转时间为 13.4，平均加权周转时间为 7.26。

RR 的平均周转时间为 9.2，平均加权周转时间为 2.84。

SJF 的平均周转时间为 7，平均加权周转时间为 1.74。

非剥夺式优先级调度算法的平均周转时间为 12，平均加权周转时间为 6.36。

30. (1)具有两道作业的批处理系统，内存只存放两道作业，它们采用抢占式优先级调度算法竞争 CPU,而将作业调入内存采用的是短作业优先调度。8：O0,作业 1 到来，此时内存和 CPU 空闲，作业 1 进入内存并占用 CPU:8:20,作业 2 到来，内存仍有一个位置空闲，因此将作业 2 调入内存，又由于作业 2 的优先数高，相应的进程抢占 CPU,在此期间 8：30 作业 3 到来，但内存此时已无空闲，因此等待。直至 8：50，作业 2 执行完毕，此时作业 3、4 竞争空出的一道内存空间，作业 4 的运行时间短，因此先调入，但它的优先数低于作业 1，因此作业 1 先执行。到 9：10 时，作业 1 执行完毕，再将作业 3 调入内存，且由于作业 3 的优先数高而占用 CPU。所有作业进入内存的时间及结束的时间见下表。

作业	到达时间	运行时间	优先数	进入内存时间	结束时间	周转时间
1	8:00	40min	5	8:00	9:10	70min
2	8:20	30min	3	8:20	8:50	30min
3	8:30	50min	4	9:10	10:00	90min
4	8:50	20min	6	8:50	10:20	90min

(2)平均周转时间为(70+30+90+90)/4=70min。

31.

作业的响应比可表示为

$$响应比=\frac{等待时间+要求服务时间}{要求服务时间}$$

在时刻 8:00，系统中只有一个作业 J_1，因此系统将它投入运行。在 J_1 完成（10:00）时，J_2, J_3, J_4 的响应比分别为(90 + 40)/40, (60 + 25)/25, (30 + 30)/30，即 3.25, 3.4.2，因此应先将 J_3 投入运行。在 J_3 完成（10:25）时，J_2, J_4 的响应比分别为(115 + 40)/40, (55 + 30)/30，即 3.875, 2.83，因此应先将 J_2 投入运行，待它运行完毕时（11:05），再将 J_4 投入运行，J_4 的结束时间为 11:35。

可见作业的执行次序为 J_1, J_3, J_2, J_4，各作业的运行情况见下表，它们的周转时间分别为 120min, 155min, 85min, 125min，平均周转时间为 121.25min。

作业号	提交时间	开始时间	执行时间	结束时间	周转时间
1	8:00	8:00	2h	10:00	120min
2	8:30	10:25	40min	11:05	155min
3	9:00	10:00	25min	10:25	85min
4	9:30	11:05	30min	11:35	125min

44.我校应用题常考题，需要格外注意！！！

本题是一个典型的利用信号量实现前驱关系的同步问题。

```
Semaphore a=b=c=d=e=f=g=h=0;     //定义进程执行顺序的信号量
CoBegin
process P1(){
    执行 P1 的任务;
    V(a);                  //实现先 P1 后 P2 的同步关系
    V(b);                  //实现先 P1 后 P3 的同步关系
}
process P2(){
    P(a);                  //检查 P1 是否已运行完成
    执行 P2 的任务;
    V(c);                  //实现先 P2 后 P4 的同步关系
    V(d);                  //实现先 P2 后 P5 的同步关系
}
process P3(){
    P(b);                  //检查 P1 是否已运行完成
    执行 P3 的任务;
    V(e);                  //实现先 P3 后 P4 的同步关系
    V(f);                  //实现先 P3 后 P5 的同步关系
}
process P4(){
    P(c);                  //检查 P2 是否已运行完成
    P(e);                  //检查 P3 是否已运行完成
    执行 P4 的任务;
    V(g);                  //实现先 P4 后 P6 的同步关系
}
process P5(){
    P(d);                  //检查 P2 是否已运行完成
    P(f);                  //检查 P3 是否已运行完成
    执行 P5 的任务;
    V(h);                  //实现先 P5 后 P6 的同步关系
}
process P6(){
    P(g);                  //检查 P4 是否已运行完成
    P(h);                  //检查 P5 是否已运行完成
    执行 P6 的任务;
CoEnd
```

45.因为去年没考，本类型的题为今年大题重点考点！！！

思路：将独木桥的两个方向分别标记为 A、B;并用整形变量 countA、countB 分别表示 A、B 方向上独木桥的行人数，初值为 O;再设置三个初值都为 1 的互斥信号量：SA 用来实现 countA 的互斥访问，SB 用来实现 countB 的互斥访问，mutex 用来实现两个方向的行人对独木桥的互斥使用。int countA=0,countB=0;semaphore SA=1,SB=1,mutex=1;

```
Process A()
{
   P(SA);
   if(countA==0)
      P(mutex);
   countA=countA+l;
   V(SA);
   过独木桥；
   P(SA);
   countA=countA-1;
   if(countA==0)
     V(mutex):
   V(SA);
}
```

```
Process B()
{
   P(SB);
   if(countB==0)
      P(mutex);
   countB=countB+l;
   V(SB);
   过独木桥；
   P(SB);
   countB=countB-1;
   if(countB==0)
      V(mutex):
   V(SB);
}
```

46.

本题是生产者-消费者问题的变体，生产者“车间 A”和消费者“装配车间”共享缓冲区“货架 F1”:生产者“车间 B”和消费者“装配车间”共享缓冲区“货架 2”。因此，可为它们设置 6 个信号量：empty1 对应货架 F1 上的空闲空间，初值为 10：fulll1 对应货架 F1 上面的 A 产品，初值为 0：empty2 对应货架 F2 上的空闲空间，初值为 10：full2 对应货架 F2 上面的 B 产品，初值为 0：mutex1 用于互斥地访问货架 F1,初值为 1：mutex2 用于互斥地访问货架 F2,初值为 1。

A 车间的工作过程可描述为：

```
while(1){
    生产一个产品 A;
    P(empty1);                    //判断货架 F1 是否有空
    P(mutex1);                    //互斥访问货架 F1
    将产品 A 存放到货架 F1 上;
    V(mutex1);                    //释放货架 F1
    V(full1);                     //货架 F1 上的零件 A 的个数加 1
}
```

B 车间的工作过程可描述为：

```
while(1){
    生产一个产品 B;
    P(empty2);                    //判断货架 F2 是否有空
    P(mutex2);                    //互斥访问货架 F2
    将产品 B 存放到货架 F2 上;
    V(mutex2);                    //释放货架 F2
    V(full2);                     //货架 F2 上的零件 B 的个数加 1
}
```

47.从井中取水并放入水缸是一个连续的动作，可视为一个进程：从缸中取水可视为另一个进程。设水井和水缸为临界资源，引入 well 和 vat:三个水桶无论是从井中取水还是将水倒入水缸都是一次一个，应该给它们一个信号量 pal,抢不到水桶的进程只好等待。水缸满时，不可以再放水，设置 empty 信号量来控制入水量：水缸空时，不可以取水，设置 full 信号量来控制。本题需要设 置 5 个信号量来进行控制：

```
semaphore well=1;                    //用于互斥地访问水井
semaphore vat=1;                     //用于互斥地访问水缸
semaphore empty=10;                  //用于表示水缸中剩余空间能容纳的水的桶数
semaphore full=0;                    //表示水缸中的水的桶数
semaphore pail=3;                    //表示有多少个水桶可以用，初值为 3
//老和尚
while(1){
    P(full);
    P(pail);
    P(vat);
    从水缸中打一桶水;
    V(vat);
    V(empty);
    喝水;
    V(pail);
}
//小和尚
while(1){
    P(empty);
    P(pail);
    P(well);
    从井中打一桶水;
    V(well);
    P(vat);
    将水倒入水缸中;
    V(vat);
    V(full);
    V(pail);
}
```

50. 选 择 填 空 重 点 考 点 ！ ！ ！ 。

破坏互斥条件：某些资源只能被互斥访问，并且某些情况下必须保护互斥性

破坏不剥夺条件
- 释放已经占有的资源
- 特点：增加系统开销 实现复杂 降低吞吐量
- 用于状态易于保存和恢复的数据（CPU的寄存器及内存资源）

破坏请求并保持条件
- 一次性申请完所需要的全部资源　又称静态分配法
- 特点：实现简单，但是资源被严重浪费，甚至可能导致进程饥饿

破坏循环等待条件
- 采用顺序资源法，对进程进行顺序推荐
- 特点：进程编号必须稳定，可能会导致资源浪费，并且不利于用户编程

63. 应 用 题 重 点 考 点 ！ ！ ！ 。

1）

$$\text{Need} = \text{Max} - \text{Allocation} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 7 & 5 & 0 \\ 2 & 3 & 5 & 6 \\ 0 & 6 & 5 & 6 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 0 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 6 & 4 & 2 \end{bmatrix}$$

2）Work 向量初始化值 ＝Available(1, 5, 2, 0)。

系统安全性分析：

资源情况 / 进程	Work				Need				Allocation				Work + Allocation			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_0	1	5	2	0	0	0	0	0	0	0	1	2	1	5	3	2
P_2	1	5	3	2	1	0	0	2	1	3	5	4	2	8	8	6
P_1	2	8	8	6	0	7	5	0	1	0	0	0	3	8	8	6
P_3	3	8	8	6	0	6	4	2	0	0	1	4	3	8	9	10

因为存在一个安全序列<P_0, P_2, P_1, P_3>，所以系统处于安全状态。

3）$\text{Request}_1(0, 4, 2, 0) < \text{Need}_1(0, 7, 5, 0)$

$\text{Request}_1(0, 4, 2, 0) < \text{Available}(1, 5, 2, 0)$

假设先试着满足进程 P_1 的这个请求，则 Available 变为(1, 1, 0, 0)。

系统状态变化见下表：

资源情况 / 进程	Max				Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	0	0	0	0	1	1	0	0
P_1	1	7	5	0	1	4	2	0	0	3	3	0				
P_2	2	3	5	6	1	3	5	4	1	0	0	2				
P_3	0	6	5	6	0	0	1	4	0	6	4	2				

再对系统进行安全性分析，见下表：

资源情况 / 进程	Work				Need				Allocation				Work + Allocation			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_0	1	1	0	0	0	0	0	0	0	0	1	2	1	1	1	2
P_2	1	1	1	2	1	0	0	2	1	3	5	4	2	4	6	6
P_1	2	4	6	6	0	3	3	0	1	4	2	0	3	8	8	6
P_3	3	8	8	6	0	6	4	2	0	0	1	4	3	8	9	10

因为存在一个安全序列<P_0, P_2, P_1, P_3>，所以系统仍处于安全状态。所以进程 P_1 的这个请求应该马上被满足。

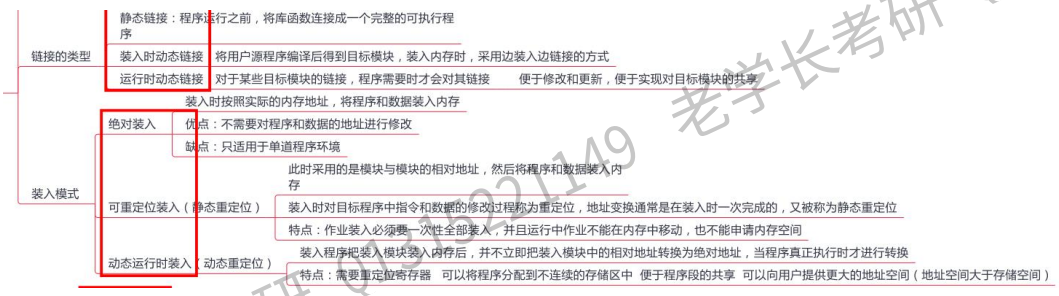
第三章内存管理

1-3 DCB　　5-10 ABD(CBBB)BD　　11-15 DDCBA

19-25 CBADCBC　　26-31 CBBBCA

4. 选择填空重要考点！！！

编译、链接、装入



16.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 位为虚页号；主存为 16 个页面，即物理地址高 4 位为物理块号。

逻辑地址 0AC5H 转换为二进制是 0001010011000101B,虚页号为 2(00010B),映射至物理块号 4，因此系统访问物理地址 12C5H(01001011000101B)。

逻辑地址 1AC5H 转换为二进制是 0011010011000101B,虚页号为 6(00110B),不在页面映射表中，会产生缺页中断，系统进行缺页中断处理。

逻辑地址 3AC5H 转换为二进制是 0111010011000101B,页号为 14，而该用户程序只有 10 页，因此系统产生越界中断。

17.此类题型应用题重点考点！！！

(1)页面的大小为 64/16)KB=4KB,该进程共有 4 页，所以该进程的总长度为 4×4KB=16KB。

(2)页面大小为 4KB,因此低 12 位为页内偏移地址：主存分为 16 块，因此内存物理地址高 4 位为主存块号。

页号为 0 的页面被装入主存的第 9 块，因此该地址在内存中的始址为 1001000000000000B,即 9000H:

页号为 1 的页面被装入主存的第 0 块，因此该地址在内存中的始址为 0000000000000000B,即 0000H。

页号为 2 的页面被装入主存的第 1 块，因此该地址在内存中的始址为 0001000000000000,即 1000H。

页号为 3 的页面被装入主存的第 14 块，因此该地址在内存中的始址为 1110000000000000,即 E000H。

(3)

逻辑地址为 0,0)，因此内存地址为(9,0)=1001000000000000B,即 9000H。

逻辑地址为(1,72)，因此内存地址为(0,72)=0000000001001000B,即 0048H。

逻辑地址为(2,1023)，因此内存地址为(1,1023)=0001001111111111，即 13FFH。

逻辑地址为(3,99)，因此内存地址为(14,99)=1110000001100011，即 E063H。

18.此为我校高频应用题考点！！！

1）在页式存储管理中，访问指令或数据时，首先要访问内存中的页表，查找到指令或数据所在页面对应的页表项，然后根据页表项查找访问指令或数据所在的内存页面。需要访问内存 2 次。

段式存储管理同理，需要访问内存 2 次。

段页式存储管理，首先要访问内存中的段表，然后访问内存中的页表，最后访问指令或数据所在的内存页面，需要访问内存 3 次。

对于比较复杂的情况，如多级页表，若页表划分为 N 级，则需要访问内存 $N+1$ 次。若系统中有快表，则在快表命中时，只需要访问内存 1 次。

2）按 1）中的访问过程分析，有效存取时间为

$$(0.2+1)\times 85\% + (0.2+1+1)\times(1-85\%) = 1.35\mu s$$

3）同理可计算得

$$(0.2+1)\times 50\% + (0.2+1+1)\times(1-50\%) = 1.7\mu s$$

从结果可以看出，快表的命中率对访存时间影响非常大。当命中率从 85%降低到 50%时，有效存取时间增加 0.35μs。因此在页式存储系统中，应尽可能地提高快表的命中率，从而提高系统效率。

32.此为我校高频应用题考点！！！

(1) 根据页面走向，使用最佳置换算法时，页面置换情况见下表。

物理块数为 3 时：

走向	4	3	2	1	4	3	5	4	3	2	1	5
块 1	4	4	4	4	4	4	4	4	4	2	2	2
块 2		3	3	3	3	3	3	3	3	3	1	1
块 3			2	1	1	1	5	5	5	5	5	5
缺页	√	√	√	√			√			√	√	

缺页率为 7/12。

物理块数为 4 时：

走向	4	3	2	1	4	3	5	4	3	2	1	5
块 1	4	4	4	4	4	4	4	4	4	4	1	1
块 2		3	3	3	3	3	3	3	3	3	3	3
块 3			2	2	2	2	2	2	2	2	2	2
块 4				1	1	1	5	5	5	5	5	5
缺页	√	√	√	√			√				√	

缺页率为 6/12。

由上述结果可以看出，增加分配作业的内存块数可以降低缺页率。

(2) 根据页面走向，使用先进先出页面淘汰算法时，页面置换情况见下表。

物理块数为 3 时：

走向	4	3	2	1	4	3	5	4	3	2	1	5
块 1	4	4	4	1	1	1	5	5	5	5	5	
块 2		3	3	3	4	4	4	4	4	2	2	
块 3			2	2	2	3	3	3	3	3	1	
缺页	√	√	√	√	√	√	√			√	√	

缺页率为 9/12。

物理块数为 4 时：

走向	4	3	2	1	4	3	5	4	3	2	1	5
块 1	4	4	4	4	4	4	5	5	5	5	1	1
块 2		3	3	3	3	3	3	4	4	4	4	5
块 3			2	2	2	2	2	2	3	3	3	3
块 4				1	1	1	1	1	1	2	2	2
缺页	√	√	√	√			√	√	√	√	√	√

缺页率为 10/12。

由上述结果可以看出，对先进先出算法而言，增加分配作业的内存块数反而使缺页率上升，即出现 Belady 现象。

(3) 根据页面走向，使用最近最久未使用页面淘汰算法时，页面置换情况见下表。

物理块数为 3 时：

走向	4	3	2	1	4	3	5	4	3	2	1	5
块 1	4	4	4	1	1	1	5	5	5	2	2	2
块 2		3	3	3	4	4	4	4	4	4	1	1
块 3			2	2	2	3	3	3	3	3	3	5
缺页	√	√	√	√	√	√	√			√	√	√

缺页率为 10/12。

物理块数为 4 时：

走向	4	3	2	1	4	3	5	4	3	2	1	5
块 1	4	4	4	4	4	4	4	4	4	4	4	5
块 2		3	3	3	3	3	3	3	3	3	3	3
块 3			2	2	2	2	5	5	5	5	1	1
块 4				1	1	1	1	1	1	2	2	2
缺页	√	√	√	√			√			√	√	√

缺页率为 8/12。

由上述结果可以看出，增加分配作业的内存块数可以降低缺页率。

33.此为拓展题，最好还是要会！！！

在缺页中断处理完成，调入请求页面后，还需 1μs 的存取访问，即

(1)当未缺页时，直接访问内存，用时 1μs。

(2)当缺页时，若未修改，则用时 8μs+1μs。

(3)当缺页时，而且修改了，则用时 20μs+1μs。

设最大缺页中断率为 p,有(1-p)×1μs+(1-70%)×p×(1μs+8μs)+70%×p×(1μs+20μs)=2μs,即 1μs+(1-70%)×p×8μus+70%×p×20us=2μs,解得 p≈0.061=6.1%。

34.

(1)页面大小为 4KB,因此页内偏移为 12 位。系统采用 48 位虚拟地址，因此虚页号为 48-12=36 位。采用多级页表时，最高级页表项不能超出一页大小：每页能容纳的页表项数为 4KB/8B=512=2^9，36/9=4，因此应采用 4 级页表，最高级页表项正好占据一页空间。

(2)系统进行页面访问操作时，首先读取页面对应的页表项，有 98%的概率可以在 TLB 中直接读取到，然后进行地址转换，访问内存读取页面：若 TLB 未命中，则要通过一次内存访问来读取页表项。页面平均访问时间为 98%×(10+100)+(1-98%)×(10+100+100)=112ns

(3)二级页表的平均访问时间计算同理：

98%×(10+100)+(1-98%)×(10+100+100+100)=114ns

(4)设快表命中率为 p,则应满足 p×(10+100)+(1-p)×(10+100+100+100)≤120ns 解得 p≥95%。

(5)系统采用 48 位虚拟地址，每段最大为 4GB,因此段内地址为 32 位，段号为 48-32=16 位。每个用户最多可以有 2^{16}段。段内采用页式地址，与 1)中计算同理，(32-12)/9，取上整为 3，因此段内应采用 3 级页表。

35.此为我校高频应用题考点！！！

(1)FFO 算法：最先进入的页帧号应最先替换，因此访问虚页 4 发生缺页时，应置换 3 号页帧中的 3 号虚页，因为它是最先进入存储器的。

LRU 算法：应置换 1 号页帧中的 1 号虚页，因为它是最久未被访问和修改过的，又是最先进入存储器的。改进型 CLOCK 算法：第一轮扫描淘汰访问位和修改位都为 0 的页面，因此淘汰 1 号页面。

(2)采用 LRU 算法时缺页情况如下表，缺页次数为 3 次。

页访问串	当前状态	4	0	0	0	2	4	2	1	0	3	2
标记		*							*		*	
M_1	2	2	2	2	2	2	2	2	2	2	2	2
M_2	1	4	4	4	4	4	4	4	4	4	3	3
M_3	0	0	0	0	0	0	0	0	0	0	0	0
M_4	3	3	3	3	3	3	3	3	1	1	1	1

第四章文件管理

2-6 DBDC　　6-8 AAB

1.逻辑结构：顺序文件、索引文件、顺序索引文件、直接文件或散列文件

存储结构：连续分配、链接分配(显示链接、隐式链接)、索引分配

不同：逻辑结构强调的是文件之间的逻辑关系，而存储结构强调的是文件具体在计算机上是怎样存储的(注意比较索引在逻辑结构和存储结构中的不同)

9.此为我校文件管理最重要知识点，连续几年都在考，务必彻底掌握！！！

第一问要计算混合索引结构的寻址空间大小：第二问只要计算出存储该文件索引块的大小，然后加上该文件本身的大小即可。

(1)物理块大小为 4KB,数据大小为 4B,则每个物理块可存储的地址数为 4KB/4B=1024。

最大文件的物理块数可达 10+1024+10242+10243，每个物理块大小为 4KB,因此总长度为$(10+1024+1024^2+1024^3)\times 4KB=40KB+4MB+4GB+4TB$

这个文件系统允许的最大文件长度是 4TB+4GB+4MB+40KB,约为 4TB。

(2)占用空间分为文件实际大小和索引项大小，文件大小为 2GB,从 1)中的计算知，需要使用到二次间接索引项。该文件占用 2GB/4KB=512×1024 个数据块。

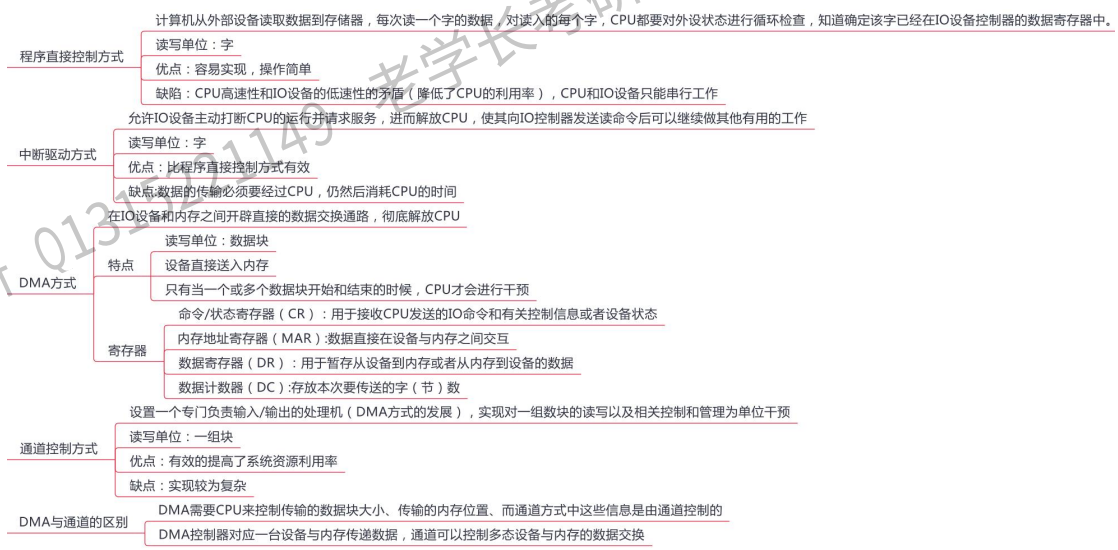
一次间接索引项使用 1 个间接索引块，二次间接索引项使用 1+「(512×1024-10-1024)/1024≈512 个间接索引块（最左的 1 表示二次间址块)，所以间接索引块所占空间大小为(1+512)×4KB=2MB+4KB

另外每个文件使用的 i node 数据结构占 13×4B=52B,因此该文件实际占用磁盘空间大小为 2GB+2MB+4KB+52B.

第五章 IO 管理

3-5 BCC　　6-9 CDAA　　11-15 ADABB　　16-18 BCA

1.



2.用户层 IO 软件、设备独立性软件、设备驱动程序、中断处理程序、硬件设备
(一定要区分前后顺序，此顺序为从上到下，从软件到硬件)

10.预测选择或填空考点，最好还是要理解！！！

寻道时间：移动磁头寻找柱面所需要的时间，此部分最耗费时间。

①可以选用恰当的磁盘调度算法解决。

②采用(柱面号、盘面号、扇区号)的磁盘地址结构

延迟时间：寻找到柱面后，找到对应的扇区所消耗的时间。

①在不同的盘面同一磁道的扇区号使用错位命名

②在相同的盘面同一磁道内扇区号采用交替编号

传输时间：从磁盘读出或写入的时间

①升级硬件

19.此为我校应用题高频考点！！！

(1)FCFS:移动磁道的顺序为 345,123,874,692,475,105,376。磁头臂必须移过的磁道的数目为 222+751+182+217+370+271=2013。

(2)SSTF:移动磁道的顺序为 345,376,475,692,874,123,105。磁头臂必须移过的磁道的数目为 31+99+217+182+751+18=1298。

注意，磁头臂必须移过的磁道的数目之和的计算没有必要像上面一样对 31,99.217,182.751,18 求和，仔细的读者会发现：从 345 到 874 是一路递增的，接着从 874 到 105 是一路递减的。所以仅需计算(874-345)+(874-105)=1298。这种方法是不是要比上面得出 6 个数后再计算它们的和要快捷一些？若之前未注意到此法，相信聪明的读者会马上回顾刚做完的(1)，并会仔细观察以下几问的“规律”，进而总结出自己的思路。

3)SCAN:移动磁道的顺序为 345,123,105,0,376,475,692,874。磁头臂必须移过的磁道的数目为 222+18+105+376+99+217+182=1219。

(4)L00K:移动磁道的顺序为 345,123,105,376,475,692,874。磁头臂必须移过的磁道的数目为 222+18+271+99+217+182=1009。

(5)C-SCAN:移动磁道的顺序为 345,123,105,0,999,874,692,475,376。磁头臂必须移过的磁道的数目为 222+18+105+999+125+182+217+99=1967。

(6)C-L00K:移动磁道的顺序为 345,123,105,874,692,475,376。磁头臂必须移过的磁道的数目为 222+18+769+182+217+99=1507。