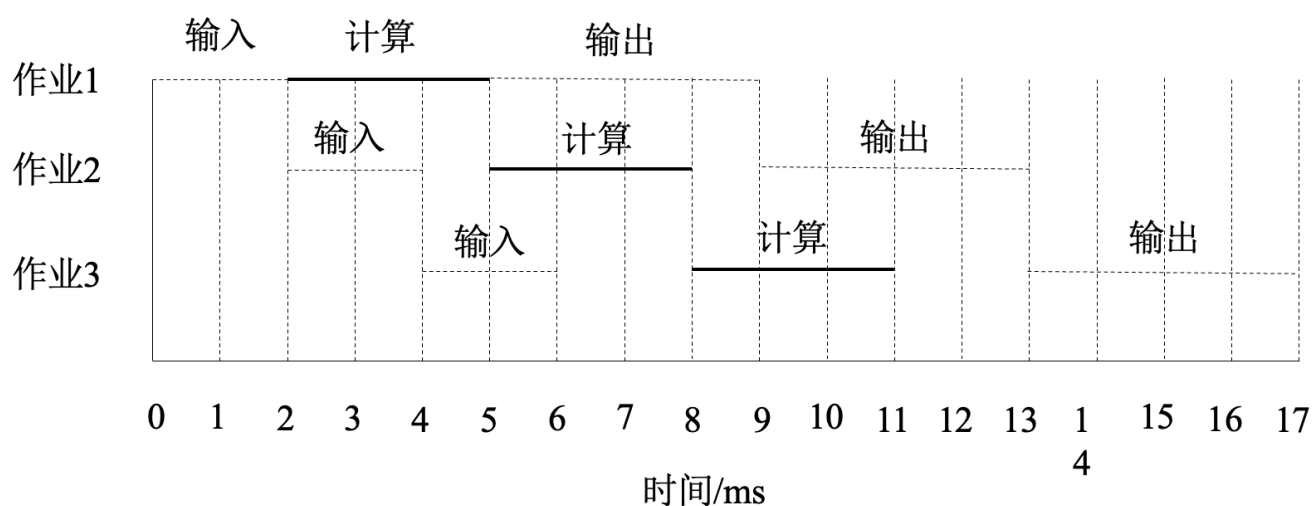


# 第一章

- 1、**OS**对系统必不可少。
- 2、从用户的角度看，**OS是用户与计算机硬件系统之间的接口**
- 3、配置了OS的计算机是一台比原来的物理计算机功能更加强大的计算机，这样的计算机只是一台逻辑上的计算机，称为**虚拟计算机**。
- 4、与单道程序系统相比，多道程序系统的优点是：**CPU利用率高，系统吞吐量大，I/O设备利用率高**
- 5、**批处理系统分为单道批处理系统和多道批处理系统**，中断技术使得多道批处理系统和I/O设备可与CPU并行工作
- 6、**分时系统允许一台主机上同时连接多台终端，多个用户可以通过各自的终端同时交互地使用计算机。**
- 7、**航空订票系统，机床控制系统，股票交易系统**等的实现最好采用实时系统平台。
- 8、OS的基本类型主要有**批处理系统、分时系统和实时系统**
- 9、并发性是指若干事件在**同一时间间隔内**发生
- 10、单处理机系统中，可并行的是：**处理机与设备，处理机与通道，设备与设备**
- 11、

某单CPU系统中有输入设备和输出设备各1台，现有3个并发执行的作业，每个作业的输入、计算和输出时间分别为2ms、3ms和4ms，且都按输入、计算和输出的顺序执行，则执行完这3个作业需要的时间最少是多少？



- 12、OS提供给应用程序的接口是**系统调用**
- 13、**整数除以0，read系统调用**，都是会导致用户进程从用户态切换到内核态的操作，因为**整数除以0会发生异常**，需要在内核态执行；系统调用也需要在内核态执行
- 14、**不同的OS**为应用程序提供的系统调用接口**不一定相同**。

15、在执行系统调用服务程序的过程中，CPU处于内核态

16、OS通过提供系统调用来避免用户程序直接访问外设

17、系统调用是OS内核为应用程序提供服务的接口

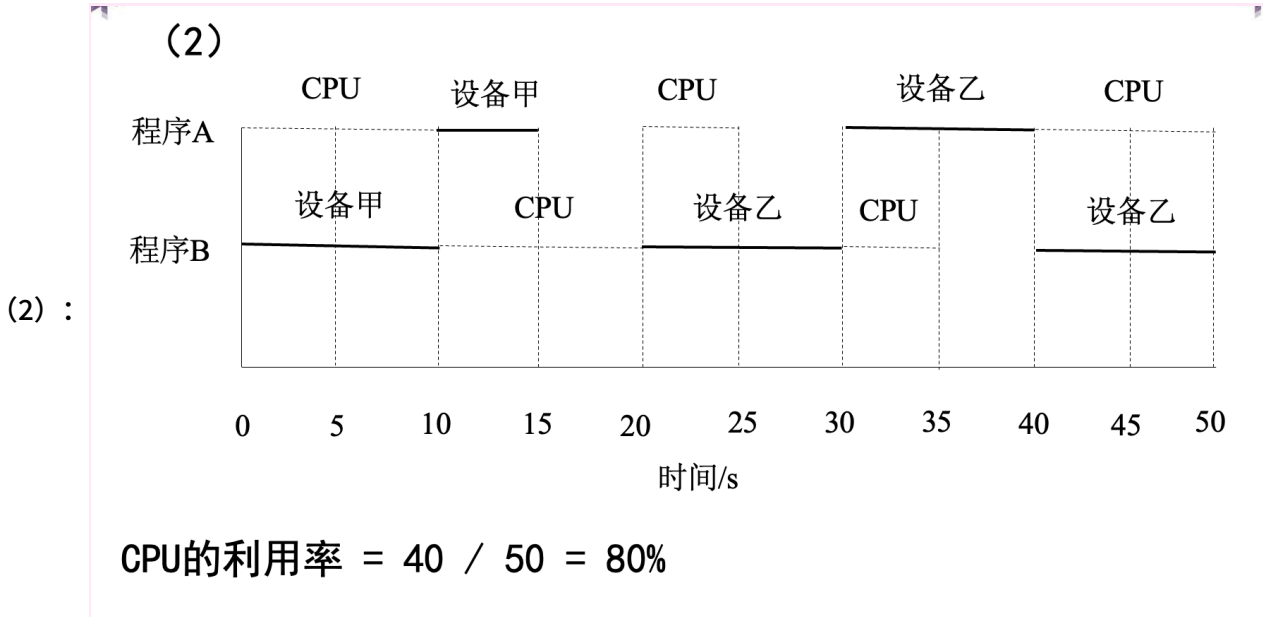
18

现有A、B两个程序，程序A按顺序使用CPU10s，使用设备甲5s，使用CPU5s，使用设备乙5s，使用CPU10s。程序B按顺序使用设备甲10s，使用CPU10s，使用设备乙5s，使用CPU5s，使用设备乙10s。试问

(1) 在顺序执行程序A和程序B的情况下，CPU的利用率是多少？

(2) 在多道程序环境下，CPU的利用率是多少？请画出A、B程序的执行过程。

(1) :  $(10+5+10+10+5) / (10+5+5+5+10+10+10+5+5+10) = 40/75 = 53.3\%$



19、在多道批处理系统中，是否并发的进程越多，资源利用率越好？为什么？

并不是并发的进程越多资源利用率越好。若系统中并发的进程过多，则会导致系统在多个进程之间频繁切换，系统的开销增大，从而降低资源利用率。

## 第二章

1、程序顺序执行时的特征：顺序性，封闭性，可再现性

2、程序并发执行时的特征：间断性，失去封闭性，不可再现性

3、程序运行时独占系统资源，只有程序本身才能改变系统资源状态，这是指程序顺序执行时的封闭性

4、进程实体由程序段、相关的数据段和PCB三部分构成

5、进程和程序的本质区别在于前者可以并发执行，后者不能并发执行

6、一般而言，每一个进程至少应处于以下三种基本状态之一：就绪状态，执行状态，阻塞状态

7、一般情况下，分时系统中处于就绪状态的进程最多。

8、在单处理机系统中，**最多只有一个进程能处于运行状态**

9、已经获得除**CPU**以外的运行所需所有资源的进程处于就绪状态

10、当一个进程**等待协作进程的一个消息**时，称其处于阻塞状态。

11、一个进程的读磁盘操作完成后，OS针对该进程必做的是**修改进程状态为就绪状态**

解析：进程申请读磁盘操作的时候，因为要等待I/O完成，将自身阻塞，进入阻塞态。当I/O操作完成之后，从阻塞进入就绪态。

12、可能导致当前进程P阻塞的事件是**进程P申请临界资源，进程P从磁盘读数据**

解析：读取资源：运行态变成阻塞态；

13、进程状态的转换中，**就绪->阻塞**是不可能发生的

14、在实时系统中，当内存资源无法满足执行紧迫任务的需求时，OS可能将正在运行的进程的状态变为**静止就绪**状态

15、进程控制块PCB**作为独立运行基本单位的标志，能实现间断性运行方式，提供进程管理所需要的信息，提供进程调度所需要的信息，实现与其它进程的同步与通信**

16、进程的状态和优先级信息存放在**PCB**中

17、OS通过**PCB**来感知进程的存在。

18、进程控制一般是由OS的内核中的**原语**来实现的

解析：进程调度是内核的进程控制原语之一

19、OS中有一组特殊的程序，它们不能被系统中断。在OS中它们称为**原语**

20、原语Creat创建一个新进程的步骤为：**申请空白PCB，为新进程分配其运行所需的资源，初始化进程控制块PCB，将新进程插入就绪队列。**

**21、生产者——消费者问题（在b站上找个网课看明白）♥**

22、

两端程序分别如下，其中，R1、R2是寄存器，counter是共享变量，且其初值为1。

程序A：R1 = counter; R1 = R1 + 1; counter = R1;

程序B：R2 = counter; R2 = R2 - 1; counter = R2;

在这两端程序对应地进程并发执行后，counter所有可能的值是什么？

解析：按这个顺序执行

```
R1 = counter; //R1 = 1
R1 = R1 + 1; //R1 = 2
counter = R1; //counter = 2
R2 = counter; //R2 = 2
R2 = R2 - 1; //R2 = 1
counter = R2; //counter = 1
结果为1;
```

解析：按这个顺序执行

```
R1 = counter; //R1 = 1
R1 = R1 + 1; //R1 = 2
R2 = counter; //R2 = 1
R2 = R2 - 1; //R2 = 0
counter = R2; //counter = 0
counter = R1; //counter = 2
结果为2;
```

解析：按这个顺序执行

```
R1 = counter; //R1 = 1
R1 = R1 + 1; //R1 = 2
R2 = counter; //R2 = 1
R2 = R2 - 1; //R2 = 0
counter = R1; //counter = 2
counter = R2; //counter = 0
结果为0;
```

23、一个访问临界资源的循环进程描述如下：

```
while (true)
{
    进入区
    临界区
    退出区
    剩余区
};
```

24、关于临界区，正确的说法是访问不同临界资源的两个进程不要求必须互斥地进入临界区

25、临界区是指并发进程中访问共享变量的程序段

26、实现临界区互斥机制所必须遵循的是：两个进程不能同时进入临界区，允许进程访问空闲的临界资源，进程等待进入临界区的时间是有限的，不能进入临界区且处于执行状态的进程立即放弃CPU

26、

```
while (TS(&lock)); 这条语句能不能改成
while (lock);
lock = true;
```

不能。如果\*lock是false，则可能出现：进程A执行完第一句后切换到进程B，这时进程A、进程B同时进入临界区

27、同步机制中，可以实现让权等待的是信号量机制

**解析：**使用信号量机制，可以防止多个线程同时访问共享资源。

ps:下面这几道题如果到了考前还不会就抓紧背，不会的题目放到牛客网去搜索，那里的程序员给的解释要比一些搜题软件要好!!!

28、设与某资源相关联的信号量初值为3，当前值为1，则该资源的可用个数和等待资源的进程数分别为1, 0

**解析：**初始值3，当前1，也就是有两个进程正在占用资源，还可以进入一个，没有进程等待。

29、若记录型信号量S的初值为17，当前值为-17，则表示有17个等待进程

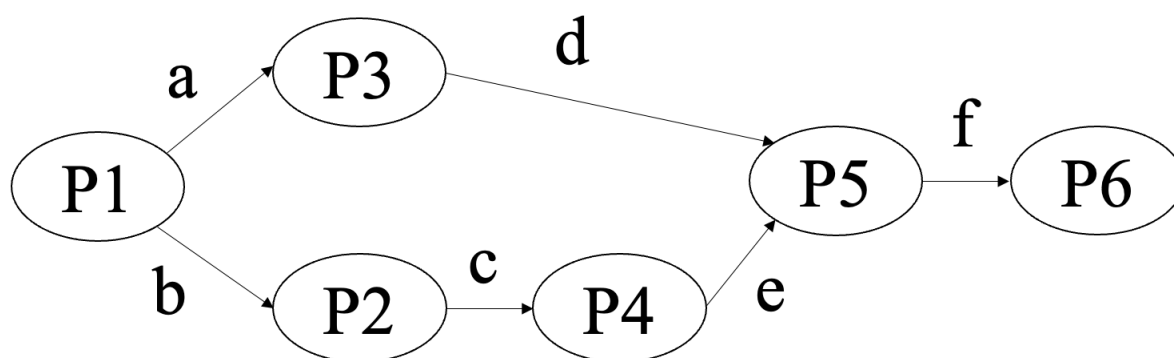
**解析：**信息量S小于0时，S的绝对值表示等待某资源的进程数量

30、如果3个进程共享一个互斥段，每次最多允许2个进程进入该互斥段，则信号量的变化范围是2, 1, 0, -1

### 31、进程的描述与控制之信号量的应用 (b站找视频)

32、

有一组相互合作的进程P1、P2、P3、P4、P5、P6，它们的执行过程须满足下图所示的同步关系，请使用信号量机制对该组进程进行同步。



```
P1(){S1; signal(a); signal(b); }
P2(){wait(b); S2; signal(c); }
P3(){wait(a); S3; signal(d); }
P4(){wait(c); S4; signal(e); }
P5(){wait(d); wait(e); S5; signal(f); }
P6(){wait(f); S6; }
main(){
    semaphore a, b, c, d, e, f;
    a = b = c = 0;
    d = e = f = 0;
    P1(); P2(); P3(); P4(); P5(); P6();
}
```

在main函数中，信号量a、b、c、d、e、f都被初始化为0。通过对信号量的等待（wait）和发送信号（signal）操作，这些进程能够按照一定的顺序同步执行。通过控制信号量的值，可以确保每个进程在满足特定条件时才能继续执行，从而实现进程同步。

33、管程是一种进程同步机制，任何时候只能有一个进程在管程中执行，管程中定义的变量只能被管程内的过程访问，管程不仅可以用于实现进程的互斥，还可以用于实现进程的同步。

34、若x是管程内的条件变量，则当执行x.wait时，所做的工作是：根据x的值判断该进程是否进入阻塞状态

### 35、生产者-消费者问题

#### 36、重中之重（要背下来这一题）

x.wait: 正在调用管程的进程因x条件需要被阻塞或挂起，则调用x.wait操作时将自己插入x条件的等待队列中，并释放管程，直到x条件变化。

x.signal 表示正在调用管程的进程发现x条件发生了变化，则调用x.signal，唤醒一个因x条件被阻塞或挂起的进程，如果没有这样的进程，继续执行原进程。

系统中有多个生产者进程和消费者进程，共享一个可以存放1000个产品信息的缓冲区（初始为空），当缓冲区未满时，生产者进程可以在其中放入一个其生产的产品信息，否则等待；当缓冲区不空时，消费者进程可以在其中取走1个产品信息，否则等待。要求1个消费者进程从缓冲区连续取走10个产品信息后，其他消费者进程才可以取产品信息。请用P、V操作或wait( )、signal( )操作实现进程间的同步。

```
item buffer[1000];
item nextp;
item nextc[10];
int in = 0, out = 0;
semaphore empty = 1000, full = 0;
//资源信号量
semaphore mutex1 = 1;
//用于生产者和消费者之间的互斥
semaphore mutex2 = 1;
//用于消费者之间的互斥
void producer(){
do{
    producer an item nextp;
    wait(empty);
    wait(mutex1);
    buffer[in] = nextp;
    in = (in + 1) % 1000;
    signal(mutex1);
    signal(full);

}while(TRUE);
void consumer(){
    do{
        wait(mutex2);
        for (i = 0; i < 10; i++){
            wait(full);
            wait(mutex1);
            nextc[i] = buffer[out];
            out = (out + 1) % n;
            signal(mutex1);
            signal(empty);
        }
        signal(mutex2);
        consumer the items in nextc;
```

```

    }
    while(TRUE);
}

```

### 37、哲学家进餐问题

有n位哲学家围坐在一张圆桌边，每位哲学家交替地就餐和思考。在圆桌中心有m个碗，每两位哲学家间有1根筷子。每位哲学家必须取到1个碗和两侧的筷子之后才能进餐，进餐完毕后将碗和筷子放回原位，继续思考。请用P、V操作或wait()、signal()操作实现进程间的同步，要求防止出现死锁现象。

```

semaphore bowl = m; //用于协调哲学家对碗的使用
semaphore chopstick[n]; //用于协调哲学家对筷子的使用
for (int i = 0; i < n; i++)
    chopstick[i] = 1;
void pi(){ //第i位哲学家的程序
    do{
        wait(bowl);
        if (i % 2 == 1){
            wait(chopstick[i]);
            wait(chopstick[(i + 1) % n]);
        }
        else{
            wait(chopstick[(i + 1) % n]);
            wait(chopstick[i]);
        }
        //eat
        signal(chopstick[i]);
        signal(chopstick[(i + 1) % n]);
        signal(bowl);
        //think
    }while(TRUE);
}

```

38、

两个进程P1、P2并发执行，并用M1、M2分别实现对两个互斥共享资源R1和R2的互斥访问。这两个进程以什么次序执行会导致死锁？在不影响程序功能的情况下，请修改算法以防止死锁。

```

void P1(){
    wait(M1);
    访问R1;
    wait(M2);
    访问R1和R2;
    signal(M1);
    signal(M2);
}

```

```
void P2(){
    wait(M2);
    访问R2;
    wait(M1);
    访问R1;
    signal(M1);
    signal(M2);
}
```

P1执行了wait(M1)，申请成功，然后访问R1，然后P2执行wait(M2)，申请成功，这时会出现死锁。

1. P1获得M1，P2获得M2。
2. P1等待M2，而M2被P2持有。
3. P2等待M1，而M1被P1持有。

所以可以这样修改P2：

```
void P2(){
    wait(M1);
    wait(M2);
    访问R2;
    访问R1;
    signal(M1);
    signal(M2);
}
```

39、关于管道通信，进程对管道进行读操作和写操作都可能被阻塞，管道存在于内存中，因此容量大小不受磁盘容量大小的限制

40、管道通信是以自然字符流为单位进行写入和读出的

41、用信箱实现进程间互通信息的通信机制要有两个通信原语，它们是发送原语和接收原语。

42、一个进程可以包含多个线程，各线程共享进程的虚拟地址空间

43、在引入线程的OS中，把线程作为调度和分派的基本单位，而把进程作为拥有资源的基本单位。

44、同一进程中的线程间切换由于有许多上下文相同而可以被简化

45、内核支持线程的调度由OS完成，在有些操作系统中，用户级线程对内核来说不可见，所以OS不会为它分配TCB，用户级线程的切换可以在用户空间完成，而内核支持线程的切换需要OS进行调度，因此用户级线程的切换效率更高。

46、不管系统是否支持线程，进程都是资源分配的基本单位

## 第三章

1、假设4个作业到达系统的时刻和运行时间如下表所示。



作业	到达时刻	运行时间
J1	0	3
J2	1	3
J3	1	2
J4	3	1

**系统在t=2时开始调度作业。**若分别采用FCFS和SJF调度算法，则选中的作业分别是

解：

**FCFS（按照作业到达的先后次序来进行调度）选中 J1 作业**

**SJF（以作业的长短来计算优先级，作业越短，其优先级越高）选中 J3 作业**

2、假设某OS以单道批处理方式运行，现有4道作业，它们进入系统的时间的及运行时间如下表所示，试采用**高响应比优先调度算法**进行调度，请问这组作业的运行顺序、平均周转时间和平均带权周转时间分别是多少？

作业号	进入系统时间	运行时间（小时）
1	7:00	2
2	7:50	0.5
3	8:00	0.1
4	8:50	0.2

解：

7:00-9:00运行作业1，周转时间2小时，带权周转时间1；

这时，作业2的响应比为 $(7/6+0.5)/0.5 = 10/3$ ；

作业3的响应比为 $(1+0.1)/0.1 = 11$

作业4的响应比为 $(1/6+0.2)/0.2 = 11/6$

所以此时应该运行作业3，

9:00-9:06运行作业3，周转时间1.1小时，带权周转时间11；

这时，作业2的响应比为 $(19/15+0.5)/0.5 = 53/15$

作业4的响应比为 $(4/15+0.2)/0.2 = 35/15$

所以此时应该运行作业2

9:06-9:36运行作业2，周转时间1.77小时，带权周转时间3.54；

9:36-9:48运行作业4，周转时间0.97小时，带权周转时间4.85；

平均周转时间 =  $(2+1.1+1.77+0.97) / 4 = 1.46$ 小时

带权平均周转时间 = (1+11+3.54+4.85) / 4 = 5.1小时

3、时钟中断是系统中特定的周期性时钟节拍，OS通过它来确定时间间隔，实现时间调节。时间片越短，进程切换的次数越多，系统开销越大，当前进程的时间片用完后，该进程的状态变为就绪态。

解析：时钟中断是系统中特定的周期性时钟节拍。操作系统通过它来确定时间间隔，实现时间的延时和任务的超时。当前进程的时间片用完后，它的状态由执行态变为就绪态

4、某系统采用抢占式SJF调度算法，下表给出了5个进程的到达时间和要求运行时间。

- (1) 请将表格填写完整；
- (2) 计算这5个进程的平均带权周转时间。

进程	到达时间	运行时间	开始运行时间	完成时间	带权周转时间
P1	0:00	4小时	0:00	7:00	7/4
P2	1:00	1小时	1:00	2:00	1
P3	3:00	2小时	3:00	5:00	1
P4	6:00	5小时	7:00	14:00	8/5
P5	8:00	2小时	8:00	10:00	1

0:00-1:00      P1 余3小时  
1:00-2:00      P2  
2:00-3:00      P1 余两小时  
3:00-5:00      P3  
5:00-7:00      P1  
7:00-8:00      P4 余4小时  
8:00-10:00    P5  
10:00-14:00    P4

平均带权周转时间 = (7/4+1+1+8/5+1) / 5 = 1.27

5、RR（时间片轮转法）调度算法，不会产生饥饿现象。

6、系统两级反馈队列调度算法进行进程调度。就绪队列Q1采用RR调度算法，时间片为10ms；就绪队列Q2采用短进程优先调度算法。系统优先调度Q1队列中的进程，当Q1位空时系统才会调度Q2中的进程；新创建的进程首先进入Q1；Q1的进程执行一个时间片后若未结束，则转入Q2。若当前Q1和Q2为空，系统依次创建进程P1、P2后即开始调度进程，P1、P2需要的CPU时间分别为30ms和20ms,则进程P1、P2在系统中的平均等待时间为多少。

解：

执行顺序：  
I.P1在Q1中执行10ms(剩余20ms)；

II.P2在Q1中执行10ms(剩余10ms);

III.P2在Q2中执行10ms(剩余0ms);

IV.P1在Q2中执行20ms(剩余0ms)。

可得，P1的等待时间为第10-30ms之间，共20ms。P2的等待时间为0-10ms直接，共10ms。平均等待时间为 $(20+10)/2=15\text{ms}$

7、5个进程P1、P2、P3、P4、P5几乎同时到达，它们预期运行时间分别为10、6、2、4、8个时间单位。各进程的优先级为分别为3、5、2、1、4（数值越大，优先级越高）。请按下列调度算法计算任务的平均周转时间。

(1) FCFS（按P1-P2-P3-P4-P5顺序）调度算法。

(2) RR调度算法，假定时间片大小为2个时间单位。

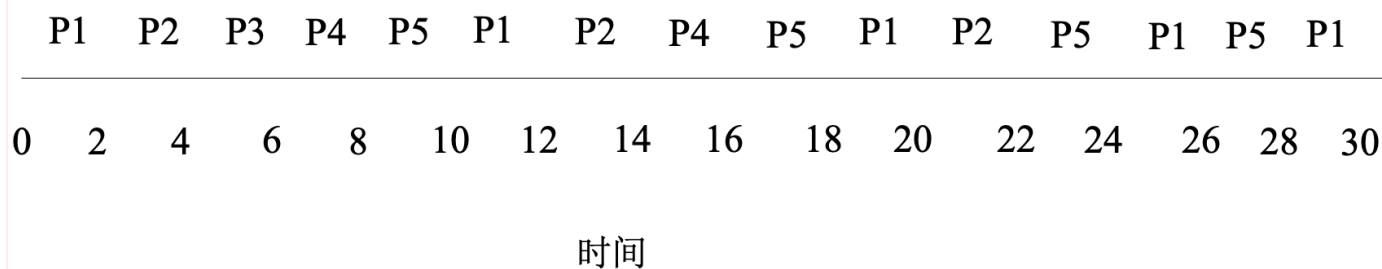
(3) 优先级调度算法。

解：

(1)：平均周转时间为 $(10+16+18+22+30)/5=19.2$ 。

(2)：看每个进程最后结束时的时间

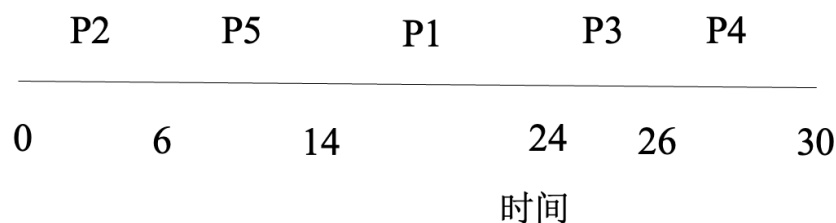
## (2) RR调度算法



平均周转时间为 $(30+22+6+16+28)/5=20.4$ 。

(3)：

## (3) 优先级调度算法



平均周转时间为 $(6+14+24+26+30)/5=20$ 。

8、某系统中有n台互斥使用的同类设备，3个并发进程分别需要3、4、5台该类设备，可确保系统不发生死锁的最小设备数为多少。

解：

$$(3-1+4-1+5-1) + 1 = 10 \text{ 台}$$

在极端状态下：

进程1(3台)：申请到2台，无法工作；

进程2(4台)：申请到3台，无法工作；

进程3(5台)：申请到4台，无法工作；

申请总数：2+3+4=9，此时若只有9台，3个进程持续申请且申请不到，造成死锁。

所以，在此时我们可以让设备要使用的资源，再多上1台，就可以使三个进程中的某一个顺利执行，从而不发生死锁。

因此，可确保系统不发生死锁的设备数  $n$  最小为10.

9、某计算机系统8台打印机，由K个进程竞争使用它们，每个进程需要3台打印机。该系统**可能会发生死锁**的K的最小值是多少。

解： 不会出现死锁的最小资源数为 $2K+1$ ；

$$2K+1 > 8 \Rightarrow K > 3.5 \Rightarrow K \text{ 为4时可能发生死锁, } K=3 \text{ 是不发生死锁}$$

**总结：用系统资源 $n-1$ 除以进程最多需要的资源数 $m-1$ 得到的结果向下取整的结果为不发生死锁的最大进程数。**

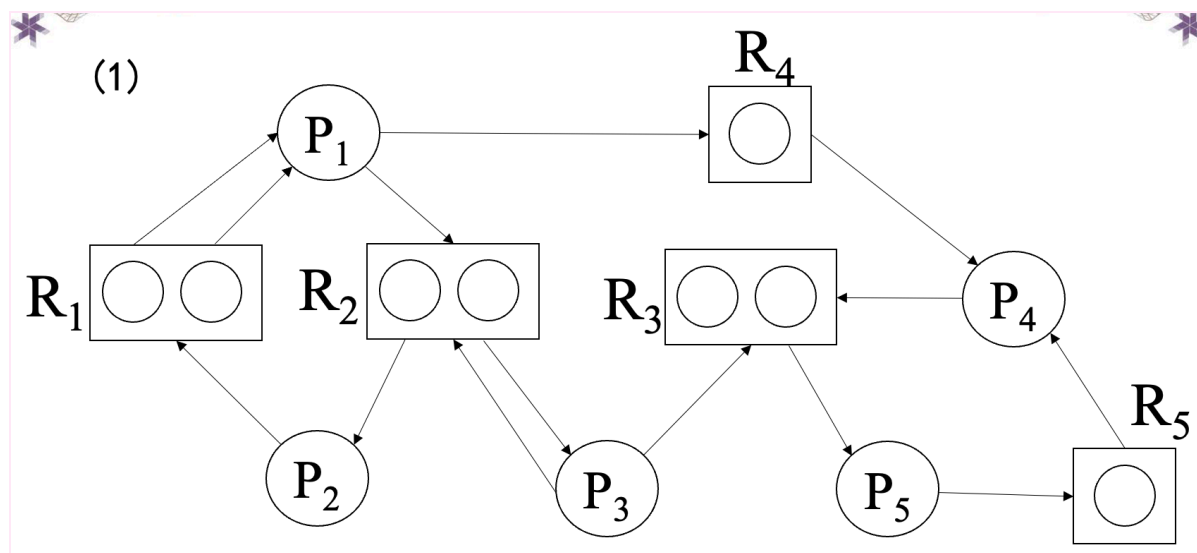
#### 10、银行家算法和安全性算法\*\* (看b站网课)

11、假设系统有5类独占资源：R1、R2、R3、R4、R5。各类资源分别有2、2、2、1、1个。系统有5个进程：

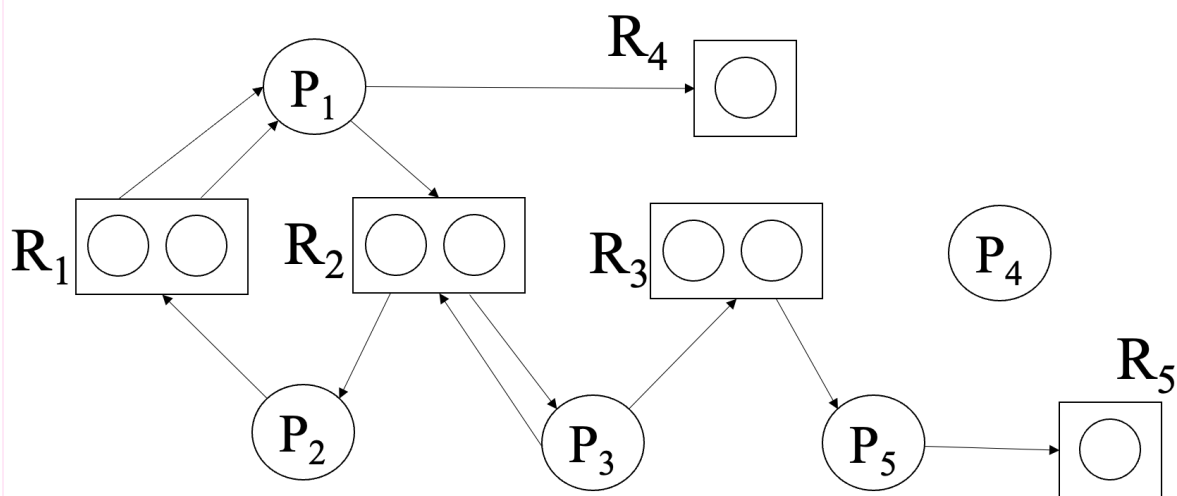
P1、P2、P3、P4、P5。其中P1已占有2个R1，且申请1个R2和1个R4；P2已占有1个R2，且申请1个R1；P3已占有1个R2，且申请1个R2和1个R3；P4已占有1个R4和1个R5，且申请1个R3；P5已占有1个R3，且申请1个R5。

(1) 画出该时刻的资源分配图；

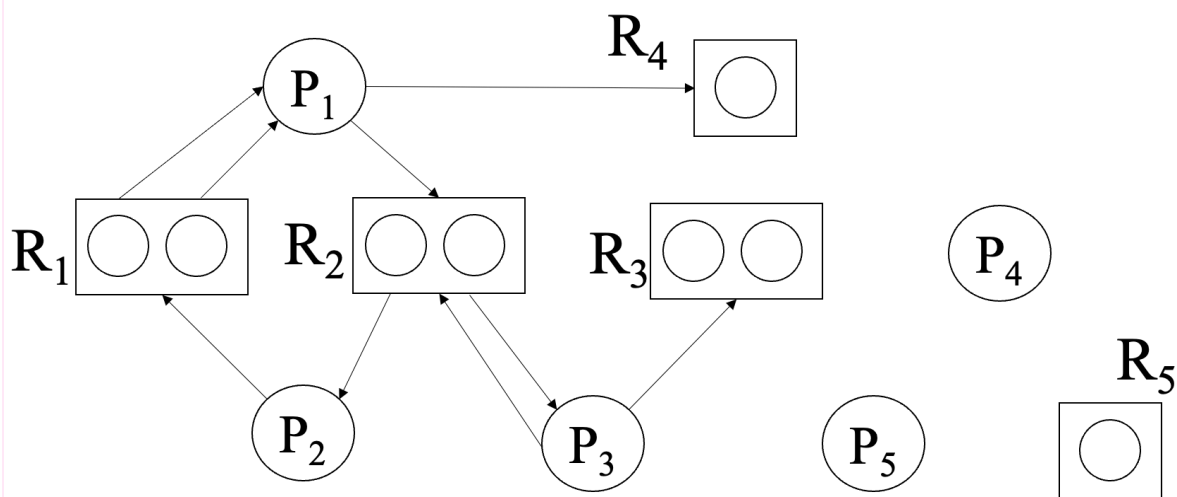
(2) 判断该时刻系统是否处于死锁状态。



(2)  $Available = (0, 0, 1, 0, 0)$ ，可以满足 $P_4$ 的要求， $P_4$ 执行完毕后，系统状态如下：



这时， $Available = (0, 0, 1, 1, 1)$ ，可以满足 $P_5$ 的要求， $P_5$ 执行完毕后，系统状态如下：



这时， $Available = (0, 0, 2, 1, 1)$ ，不能可以满足任何进程的要求，所以该资源分配图是不可完全简化的，所以系统处于死锁状态。

## 第四章

- 1、在多道程序环境中，用户程序的相对地址与装入内存后的实际物理地址不同，把相对地址转换为物理地址，这是OS的地址重定位功能。
- 2、程序链接，根据目标模块的大小和链接次序对相对地址进行修改
- 3、在程序运行前，先将一个程序的所有模块以及所需的库函数链接成一个完整的装配模块，这种链接方式称为静态链接

4、某基于动态分区存储管理的计算机，其内存容量为55MB（初始为空闲），采用**最佳适应算法**，分配和释放的顺序为：分配15MB，分配30MB，释放15MB，分配8MB，分配6MB。此时，内存中的最大空闲分区的大小是**9MB**

**最佳适应算法 (Best Fit)**：从所有空闲分区中找到**最小的一个**能够满足进程需要的空闲分区进行分配。

可重定位内存的分区分配目的是**解决碎片问题**

(1) 分配15MB，分配30MB后：

15MB	30MB	10MB
------	------	------

(2) 释放15MB后：

15MB	30MB	10MB
------	------	------

(3) 分配8MB后：

15MB	30MB	8MB	2MB
------	------	-----	-----

(4) 分配6MB后：

6MB	9MB	30MB	8MB	2MB
-----	-----	------	-----	-----

**首次适应算法 (First Fit)**：从内存的起始位置开始查找，找到第一个能够满足进程需要的空闲分区进行分配。

**最佳适应算法 (Best Fit)**：从所有空闲分区中找到最小的一个能够满足进程需要的空闲分区进行分配。

**最坏适应算法 (Worst Fit)**：从所有空闲分区中找到最大的一个能够满足进程需要的空闲分区进行分配。

**快速适应算法 (Quick Fit)**：将内存分成若干个大小相等的分区，每个分区维护一个空闲分区链表，根据进程需要的大小在相应的链表中查找空闲分区进行分配。

**分页式存储管理算法 (Paging)**：将物理内存分成若干个大小相等的页框，将进程的逻辑地址空间分成若干个大小相等的页面将页面映射到页框中，实现进程的内存分配。

**段式存储管理算法 (Segmentation)**：将进程的逻辑地址空间分成若干个大小不等的段，每个段对应一个物理内存分区，实现进程的内存分配。

5、采用动态分区算法回收内存时，如果回收分区仅与空闲分区链插入点的**前一个分区相邻接**，那么需要在空闲分区表中**修改前一个分区表项的大小**

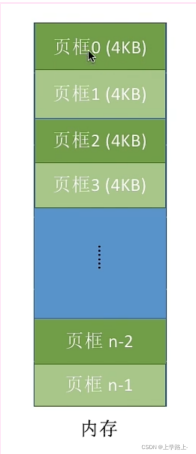
6、可重定位内存的分区分配目的是**解决碎片问题**

7、不是基于顺序搜索的动态分区分配算法的是**快速适应算法**

8、对换技术的主要作用是**提高内存利用率**

ps:插入知识：**页框，页表，页表项，页面大小，页表项 长度的理解**

**页框（物理块）：**将内存空间分成一个个大小相等的分区（页框号或物理块号从0开始）



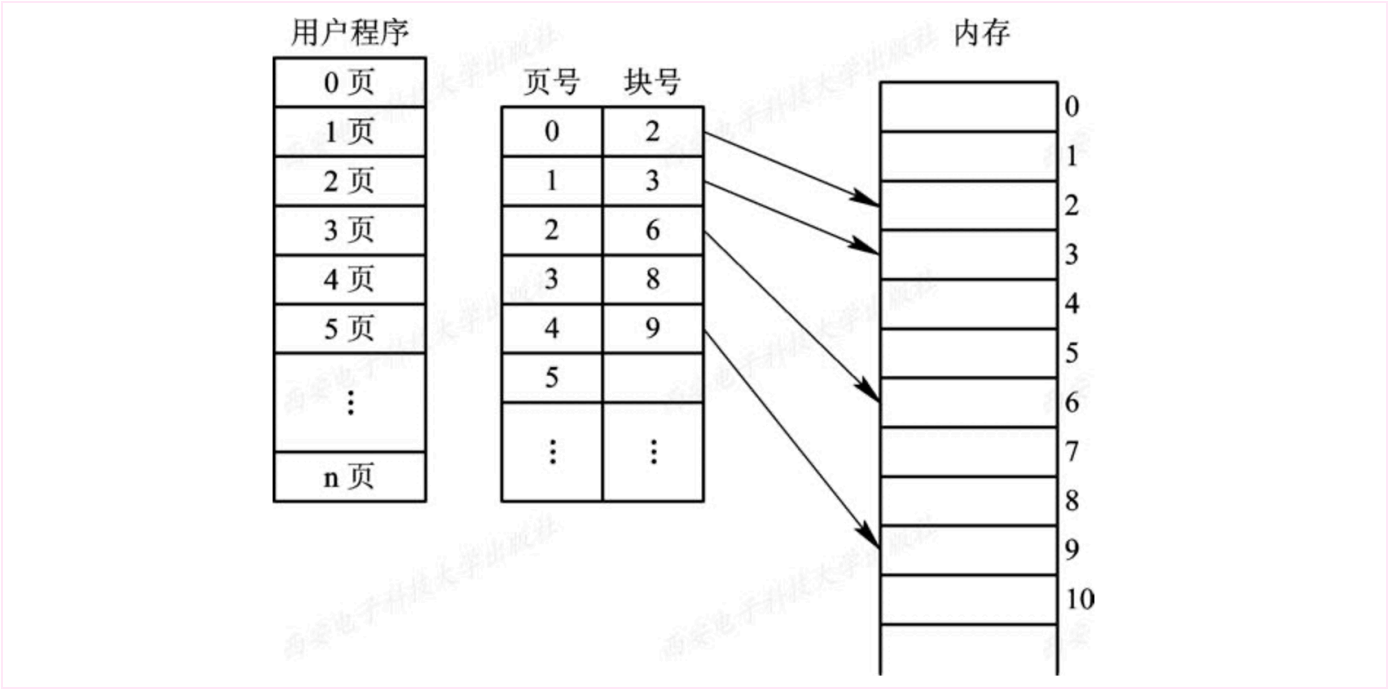
**页（页面）：**将用户进程的地址空间分为与页框大小相等的一个个区域（页号一般也从0开始）



**页面长度：**指这个页总共有几个页面或页表项

**页面大小：**指一个页面占多大的存储空间。（一般为2的12次方，也就是4KB，详细内容可以去了解分页地址中的地址结构）。

**页表：**系统为每个进程建立的页面映像表。在地址空间内的所有页（0~n），依次在页表中有一页表项，记录了相应页在内存块中对应的物理块号。



页表项：

页表项长度：指每个页表项占多大的内存空间。（每个页表项的长度是相同的）

补充：

页表在内存中占用的大小=页表长度×页表项长度；

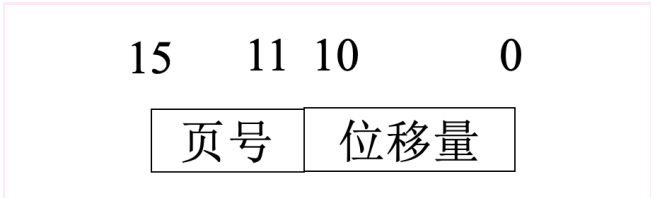
内存大小=页面长度×页面大小（物理块大小）

9、

某系统采用分页存储管理方式，拥有逻辑空间32页，每页2KB；拥有物理空间1MB。

- (1) 写出逻辑地址的格式；
- (2) 若不考虑访问权限等，则进程的页表最多有多少项？每项有多少位？
- (3) 如果物理空间减少一半，则页表结构应相应地做什么改变？

(1):该系统拥有逻辑空间 $32 = 2^5$ 页，所以逻辑地址中页号用5位来描述。每页 $2KB = 2^{11}B$ ，所以页内地址用11位来描述，逻辑地址格式如下：



(2):每个进程最多有32个页面，所以，进程的页表项最多为32项。若不考虑访问权限等，则页表中只须给出页对应的物理块号， $1MB / 2KB = 512 = 2^9$ ，所以有 $2^9$ 个块，故每个页表项有9位。

(3): $512KB / 2KB = 256 = 2^8$ ，所以有 $2^8$ 个块，故每个页表项有8位，页表中的最多项数仍是32。

10、分页系统中的地址变换过程如下：

- (1) 根据逻辑地址,计算出页号和页内偏移量；
- (2) 用页号检索页表，查找指定页面对应的块号；
- (3) 根据块号和页内偏移量，计算出物理地址：

物理地址 = 物理块号 × 页面大小 + 页内地址

11、

某分页系统的内存容量为64KB，页面大小为1KB，对一个4页大的作业，其0、1、2、3页分别被分配到内存的2、4、6、7块中。将十进制的逻辑地址1023、2500、3500转换为物理地址。

首先要计算出页号和页内地址。注意逻辑地址和物理地址都是以字节为单位。 $1KB = 1024B$ 。物理地址 = 物理块号 × 页面大小 + 页内地址。

- $1023 / 1024 = 0$ ， $1023 \% 1024 = 1023$ ，因此页号为0，页内地址为1023，对应的物理块号为2，所以物理地址为 $2 \times 1024 + 1023 = 3071$ 。
- $2500 / 1024 = 2$ ， $2500 \% 1024 = 452$ ，因此页号为2，页内地址为452，对应的物理块号为6，所以物理地址为 $6 \times 1024 + 452 = 6596$ 。
- $3500 / 1024 = 3$ ， $3500 \% 1024 = 428$ ，因此页号为3，页内地址为428，对应的物理块号为7，所以物理地址为 $7 \times 1024 + 428 = 7596$ 。



12、假设一个分页存储系统具有快表，多数活动页表项都可以存在于其中。若页表放在内存中，内存访问时间是1ns，快表的命中率是85%，快表的访问时间为0.1ns，则有效存取时间为多少？

解：EAT = 0.1ns × 85% + (0.1 + 1) × (1 - 85%) + 1 = 1.25ns.

在引入快表的分页存储管理方式中，有效访问时间的计算公式即为：

$$EAT=a \times \lambda+(t+\lambda)(1-a)+t=2t+\lambda-t \times a$$

上式中，λ表示查找快表所需要的时间，a表示命中率，t表示访问一次内存所需要的时间。

Ps:页的大小只看页内偏移量，即页内地址；虚拟空间大小为页目录号 × 页表索引；

页目录的页数 = 页目录号的总存储大小/一个页面的大小；页表的页数 = 整个页面空间的大小/一个页面的大小

第（3）小问应该先讲十六进制转化为二进制，前十位就是页目录号，发现是同一个，都为4，所以只需要访问4号页表分页即可

ps:第13题可以参考全国计算机统考2015年第46题，b站有讲解

13、某系统采用二级页表的分页存储管理方式，虚拟地址格式如下：

10位	10位	12位
页目录号	页表索引	页内地址

- (1) 页面大小是多少？进程的虚拟地址空间大小最多为多少页？
- (2) 假定页目录项和页表项均占4B，则进程的页目录和页表最多共占多少页？
- (3) 若在某指令周期内访问的虚拟地址为0100 0000H（H表示这是一个16进制数）和0111 2048H，则进行地址转换时共访问多少个页表分页？

解：

(1) 页面大小是 $2^{12}B = 4KB$ . 进程的虚拟地址空间大小最多为 $2^{10} \times 2^{10} = 2^{20}$ 页。

(2) 页目录所占页数为 $(2^{10} \times 4) / 2^{12} = 1$ ，页表所占页数为 $(2^{20} \times 4) / 2^{12} = 1024$ ，所以共占1025页。

(3) 第一个地址的前3位010转换为二进制是0000 0001 0000，前10位是0000000100 = 4（十进制）；第二个地址的前3位011转换为二进制是0000 0001 0001，前10位是0000000100 = 4（十进制），所以访问的都是4号页表分页，因此，共访问一个页表分页。

14、多级页表的优点是减少页表所占的连续内存空间

15、某计算机采用二级分页存储管理方式，页面大小为1KB，页表项大小为2B，逻辑地址空间大小为 $2^{16}$ 页，则外层页表所包含的项数至少为128

解： 一个页表分页最多只能占一个页面，所以一个页表分页占一个页面时，外层 页表的项数最少。1页可以存放 $2^{10} / 2 = 2^9$ 个页表项，所以需要 $2^{16} / 2^9 = 128$ 个 页面来存放页表项，所以外层页表的表项数至少是128

牛客解析：

在解决这个问题前，我们用类比的方式来举一个例子。

- 1. 假设有一本书，这本书每一页固定大小，最多一页有 1000 个字（**页面大小**）；
- 2. 每一页对应有一个目录中的条目，这个条目用 2 个字表示（**页表项大小**）；
- 3. 那么我 一页纸 就可以写下  $1000 / 2 = 500$  个目录条目；
- 4. 现在我整本书有 1000 页（**逻辑地址空间大小**），我们说了每一页对应一个目录条目，所以有 1000 个目录条目；
- 5. 所以我需要 1000 个目录条目 / 500 个目录条目 = 2 页（注意前面说到了，一页可写 500 个条目。所以我需要 2 个页面来保存目录条目。

16、某系统采用二级页表的分页存储管理方式，虚拟地址格式如下：

10位	10位	12位
页目录号	页表索引	页内地址

虚拟地址为2050 1225H对应的目录号和页表索引分别是什么？

解：

2050 1225H转换为二进制为：

0010 0000 0101 0000 0001 0010 0010 0101

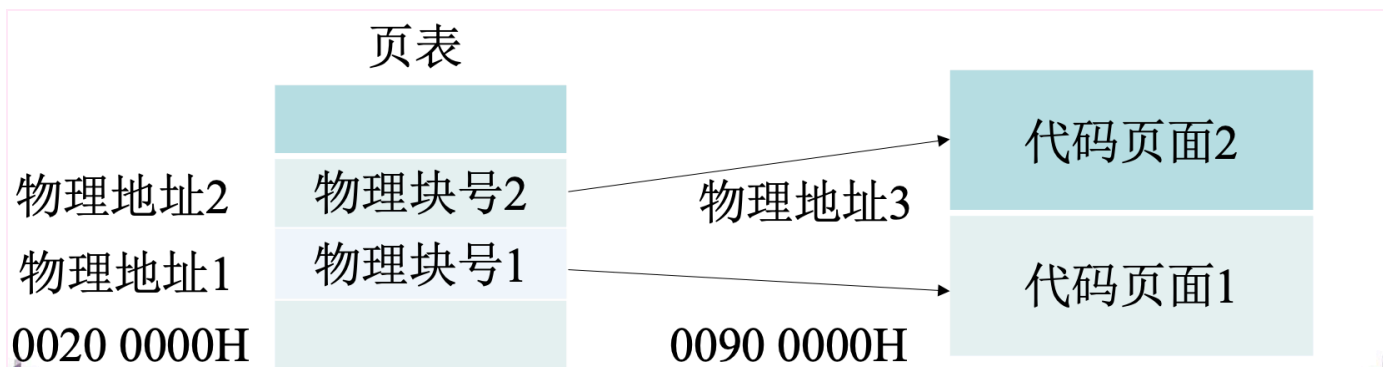
前23-32位是0010 0000 01，前面添两个0，变为0000 1000 0001 = 081H，13-22位是01 0000 0001，前面添两个0，变为0001 0000 0001 = 101H

所以答案是081H、101H

17、某系统使用一级页表的分页存储管理方式，逻辑地址和物理地址都是32位，页表项大小为4B，逻辑地址格式如下：

20位	12位
页号	页内地址

- (1) 页面大小是多少？页表最多占用多少字节？
- (2) 一个代码段的起始逻辑地址为0000 8000H，其长度为8KB，被装载到从起始物理地址为0090 0000H的连续内存空间中。页表从**内存0020 0000H开始**的物理地址处连续存放，如图所示（地址大小自下向上递增），请计算出该代码段对应的两个页表项的物理地址，以及这两个页表项中的物理块号和代码页面2的起始物理地址。



解：（1）：页面大小是 $2^{12}\text{B} = 4\text{KB}$ .页表最多占用 $2^{20} \times 4\text{B} = 4\text{MB}$

ps:第（2）问，为什么是前五位呢，因为页号是20位，每四个一位，对应前五位

（2）：0000 8000H的前5位00008转换为十进制是8，所以页号为8，物理地址1 = 起始物理地址 + 页号  $\times$  页表项大小 =  $0020\ 0000\text{H} + 8 \times 4 = 0020\ 0020\text{H}$ ，物理地址2 = 物理地址1 + 页表项大小 =  $0020\ 0020\text{H} + 4 = 0020\ 0024\text{H}$ 。由于物理块大小和页面大小相同，所以0090 000H的前5位就是物理块号，因此，物理块号1是00900H，物理块号2是00901H，物理地址3就是00901H号物理块的起始地址0090 1000H。

18、在分段管理中，以段为单位进行分配，每个段都是一个连续存储区

19、一个分段存储管理系统的地址长度为32位，其中段号占8位，则段长最大是  $2^{24}\text{B}$

解析：分段存储管理的逻辑地址分为段号和位移量两部分，段内位移的最大值就是最大段长。地址长度为32位，段号占8位，则位移量占 $32-8=24$ 位，故最大段长为 $2^{24}\text{B}$

20、采用分段存储管理不会产生内部碎片

21、在内存管理中，内存利用率高且保护和共享容易的是段页式存储管理方式

22、一个OS采用分段存储管理方式，支持的最大段长为64KB，一个进程的段表如下所示（十进制）。请问：逻辑地址47FD5H、003FFH对应的物理地址是多少？

段号	段长	段起始地址
0	512	80K
1	20K	50K
2	12K	81K
3	3K	96K
4	32K	10K

解：

（1）：64KB =  $2^{16}\text{B}$ 。逻辑地址由1个5位十六进制数，所以逻辑地址最高位数代表段号，47FD5H对应的的段号是4，段内地址是7FD5H，段长是32K =  $32768 = 2^{15}$ ，下面把它转换为十六进制：

```
32768 % 16 = 0, 32768 / 16 = 2048;
2048 % 16 = 0, 2048 / 16 = 128;
128 % 16 = 0, 128 / 16 = 8;
```

所以32768对应的十六进制是8000H，8000H > 7FD5H，没有越界。起始地址是**10K = 2800H**。物理地址 = 起始地址 + 段内地址

所以物理地址为2800H + 7FD5H = A7D5H。

(2)：003FFH对应的段号是0，段内地址是03FFH，段长是512，转换为十六进制是0200H，03FFH > 0200H，因此会产生地址越界。

## 第五章

- 1、虚拟存储管理系统的基础是程序的**局部性**原理。
- 2、虚拟存储只能基于**离散分配技术**
- 3、在请求分页存储管理中，若所需页面不在内存中，则会引起**缺页中断**
- 4、在缺页处理过程中，OS执行的操作可能是**修改页表，磁盘I/O，分配物理块**
- 5、关于进程页表的页表项，基本分页存储管理方式和请求分页存储管理方式均须设立的字段为**物理块号**
- 6、在请求分页系统中，页面分配策略与页面置换策略不能组合使用的是**固定分配，全局置换**
- 7、在请求分页存储管理中，缺页率与进程所分得的**物理块数、页面置换算法、进程页面流的走向**等因素有关
- 8、在一个请求分页系统中，系统为某进程分配了4个物理块，考虑以下页面8、1、3、6、9、8、6、1、7、0，若使用**最佳置换算法**，则访问页面9时会淘汰页面**3**

9、某系统的页面大小是1KB，某进程的大小是4.9KB，依次装入如下逻辑地址访问存储器：756、897、1044、1950、235、4000、1504、2597、2896、4501、4890、3768。

(1) 写出进程的页面访问序列。

(2) 假设系统只有2KB内存可供程序使用，假设当前时刻没有装入任何该进程的页面，若采用FIFO页面置换算法，则会发生多少次缺页中断？

(3) 若将 (2) 中的页面置换算法改为LRU页面置换算法，则会发生多少次缺页中断？

(1):  $756 / 1024 = 0$ ,  $897 / 1024 = 0$ ,  
 $1044 / 1024 = 1$ ,  $1950 / 1024 = 1$ ,  
 $235 / 1024 = 0$ ,  $4000 / 1024 = 3$ ,  
 $1504 / 1024 = 1$ ,  $2597 / 1024 = 2$ ,  
 $2896 / 1024 = 2$ ,  $4501 / 1024 = 4$ ,  
 $4890 / 1024 = 4$ ,  $3768 / 1024 = 3$ .

所以页面访问序列为0、0、1、1、0、3、1、2、2、4、4、3。

(2): 因为系统只有2KB内存可供程序使用，所以给程序分配给两个物理块。

FIFO页面置换算法：

页面序列	0	0	1	1	0	3	1	2	2	4	4	3
物理块1	0	0	0	0	0	3	3	3	3	4	4	4
物理块2			1	1	1	1	1	2	2	2	2	3
缺页	✓		✓			✓		✓		✓		✓

发生了6次缺页中断。

(3):LRU页面置换算法(最久未使用)：

页面序列	0	0	1	1	0	3	1	2	2	4	4	3
物理块1	0	0	0	0	0	0	1	1	1	4	4	4
物理块2			1	1	1	3	3	2	2	2	2	3
缺页	✓		✓			✓	✓	✓		✓		✓

发生了7次缺页中断。

10、某计算机的逻辑地址空间和物理地址空间均为64KB。若某进程最多需要6页数据存储空间，页的大小是1KB，OS采用固定分配局部置换策略为此进程分配4个物理块。进程执行到T时刻时，即将访问逻辑地址为17CAH的数据，此时页表如下所示：

页号	物理块号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	260	1

(1) 该逻辑地址对应的页号是多少？

解： 64KB = 2<sup>16</sup>B，1KB = 2<sup>10</sup>B，因此逻辑地址和物理地址的格式均为

页号/物理块号（6位）	页内地址（10位）
-------------	-----------

17CAH = 0001 0111 1100 1010，所以页号为000101 = 5

(2) 进程执行到T时刻时，即将访问逻辑地址为17CAH的数据，此时页表如下所示：

页号	物理块号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	260	1

若采用FIFO页面置换算法，则该逻辑地址对应的物理地址是多少？

解： 采用FIFO页面置换算法，则与最早调入的页面（即0号页面）进行置换，其所在的物理块号为7 = 000111，所以对应的物理地址就是把逻辑地址中的前6位改为000111后得到的地址： 0001 1111 1100 1010 = 1FCAH。

（3）若采用Clock置换算法，则逻辑地址17CAH对应的物理地址是多少？当前页面按3号页->2号页->1号页->0号页->3号页组成一个循环队列，且当前指向2号页。

页号	物理块号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	260	1

解： 从2号页面开始寻找访问位为0的页面，当指针指向的页面访问位为1，就将其改为0，遍历一周后，回到2号页面，此时2号页面的访问位为0，所以置换2号页面，其所在的物理块号为2 = 000010，所以对应的物理地址就是把逻辑地址中的前6位改为000010后得到的地址： 0000 1011 1100 1010 = 0BCAH。

11、影响请求分页系统有效（平均）访存时间的是缺页率，磁盘读写时间，内存访问时间，执行缺页处理程序的CPU时间

12、某系统采用改进型Clock页面置换算法，页表项中字段A为访问位，M为修改位。按（A，M）形式可将页分为4类（0，0），（1，0），（0，1），（1，1），则该页面置换算法淘汰页的次序为（0，0），（0，1），（1，0），（1，1）

13、请求分页管理系统中，系统采用固定分配局部置换策略，采用LRU页面置换算法。假设快表初始为空，地址转换时先访问快表，若快表未命中，再访问页表（忽略访问页表之后的快表更新时间）。

假设某进程的页表如下：

页号	物理块号	存在位
0	101H	1
1		0
2	254H	1

系统给该进程分配的物理块数是2。页面大小为4KB，一次内存访问时间是100ns，一次快表访问时间是10ns，处理一次缺页的平均时间是10<sup>8</sup>ns（已含更新快表和页表时间）。设有逻辑地址访问序列2362H、1565H、25A5H，请问：

(1) 依次访问上述3个逻辑地址，各需要多少时间？

解：4KB = 2<sup>12</sup>B，故页内地址占低的12位，页号占高的4位。

2362H的页号是2，页内地址为362H。先访问快表（10ns），未命中；再去访问页表（100ns），获得对应物理块号，其与页内地址形成物理地址，将2号页的信息装入快表中，最后根据这个物理地址访问内存（100ns）。总共需要210ns

1565H的页号是1，页内地址为565H。先访问快表（10ns），未命中；访问页表（100ns），不在内存中，产生缺页中断，中断处理时间为10<sup>8</sup>ns（已含更新快表和页表时间），此时快表已更新，访问快表（10ns）得到物理块号，其与页内地址形成物理地址，根据物理地址访问内存（100ns）。总共需要100000220ns。

25A5H的页号是2，页内地址为5A5H。先访问快表（10ns），该页信息已装入快表，得到物理块号，其与页内地址形成物理地址，根据物理地址访问内存（100ns）。总共需要110ns

(2) 基于上述访问序列，逻辑地址1565H对应的物理地址是多少？

解：访问1565H时，产生缺页中断，在处理中断时，由于采用固定分配局部置换策略，且进程的物理块数是2，内存中已经有两页，因此要淘汰1页。因为采用LRU页面置换算法，且刚刚访问了2号页，所以选择0号页置换，然后将101H号物理块分配给1号页，页内地址是565H，4K = 1000H所以物理地址是101H × 1000H + 565H = 101565H。

14、某进程访问页面的页号如下：

1,3,4,5,6,0,3,2,3,2,t, 0,4,0,3,2,9,2,1

若工作集的窗口大小为6，则在t时刻的工作集为{6,0,3,2}

解：从t时刻往前数6个页面引用，则窗口中的页面号为{6, 0, 3, 2, 3, 2}，去掉重复的3和2

15、“抖动”是指在请求分页系统中，由于页面置换算法设计不当或者进程分配的物理块数太少，刚被淘汰的页面很快又被调入内存，如此反复。

## 第六章

1、计算机I/O控制方式主要有采用轮询的可编程I/O方式、采用中断的可编程I/O方式、直接存储器访问方式和I/O通道控制方式等4种。

2、从设备的共享属性角度来讲，系统设备可划分为独占设备和共享设备

3、I/O设备一般是由执行I/O操作的机械部分和执行控制I/O的电子部件组成

4、为解决由通道不足所造成的瓶颈问题，可采取多通路技术

5、为了缓和CPU和I/O设备速度不匹配的矛盾，提高CPU和I/O设备的并行性，现代OS实现I/O设备与CPU之间的数据交换时几乎都用到了缓冲区

6、中断处理的正确流程为：

1. 保护被中断进程的CPU现场环境
2. 转入对应的中断处理子程序



3. 执行中断处理子程序
4. 恢复被中断进程的CPU现场环境

7、一般而言，设备驱动程序的功能应包括

1. 检查用户I/O的请求合法性，了解设备工作状态、传递操作控制参数并设置工作方式
2. 接收来自设备无关性软件层的命令和参数，并将其转化为与设备相关的低级操作序列
3. 根据设备状态阻塞请求进程或发出I/O命令启动设备
4. 及时响应和处理源自设备控制器的中断请求

8、就I/O控制方式而言，**直接存储器访问方式，I/O通道控制方式**支持内存和外设之间进行直接的数据传输。

9、关于I/O控制方式，**I/O通道**控制方式使对I/O操作的组织和数据的传送能最大限度地独立运行而无需处理机干预。

10、为提高OS自身的可适应性和可扩展性，现代OS通过引入**逻辑设备**的概念实现了设备独立性。

11、为了实现设备的独立性，须从如下几方面着手：

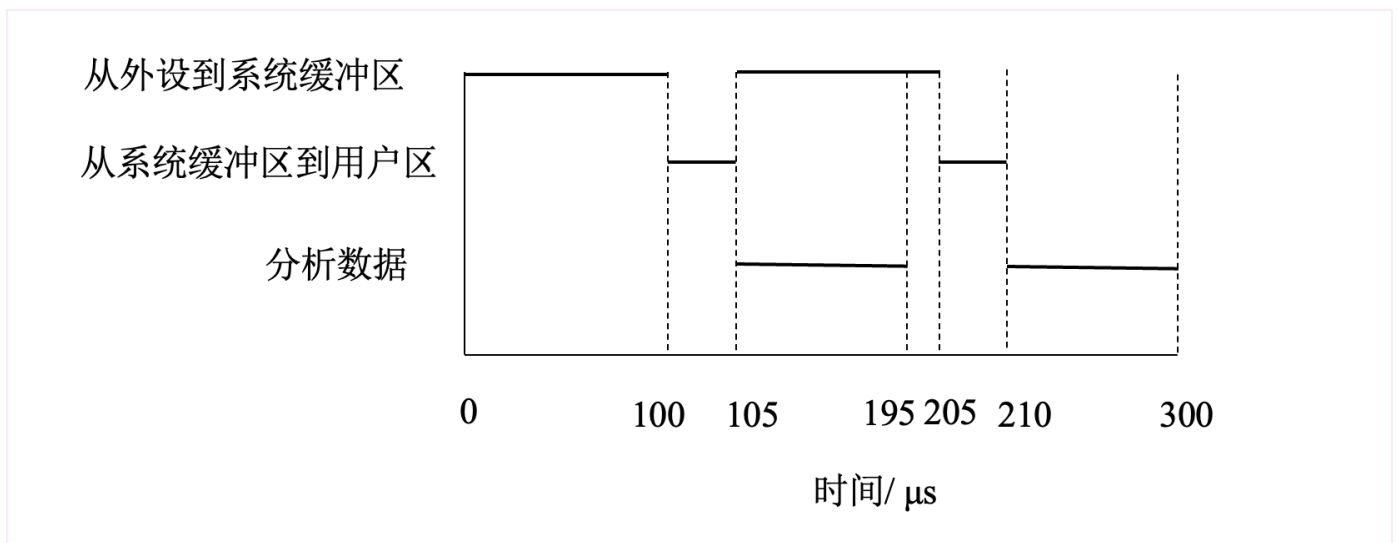
1. 引入并区分物理设备和逻辑设备这两个概念
2. 在应用程序中须使用逻辑设备名来请求和使用某类设备
3. 在应用程序中须使用物理设备名来请求和使用某类设备

12、通过硬件和软件的功能扩充，把原来独占的设备改造成若干用户所共享的设备，这种设备称为**虚拟设备**

13、关于SPOOLing技术的叙述，有**SPOOLing技术需要外存的支持，SPOOLing技术需要多道程序设计技术的支持，SPOOLing技术可以让多个作业共享一台独占设备，SPOOLing技术由井管理程序控制设备与磁盘井之间信息的交换。**

14、在设备管理中，引入缓冲的主要原因**不包括**缓和处理器和内存之间访问速度不匹配的矛盾

15、设系统缓冲区和用户工作区均采用单缓冲区，从外设读入1个数据块到系统缓冲区的时间为100  $\mu\text{s}$ ，从系统缓冲区读入1个数据块到用户工作区的时间为5  $\mu\text{s}$ ，CPU对用户工作区中的1个数据块进行分析的时间为90  $\mu\text{s}$ 。进程从外设读入并分析2个数据块的最短时间是多少？



16、用户程序发出磁盘I/O请求后，系统的处理流程是：用户程序->系统调用处理程序->设备驱动程序->中断处理程序。其中，用于计算数据所在磁盘的柱面号、磁道号、扇区号的程序是**设备驱动程序**

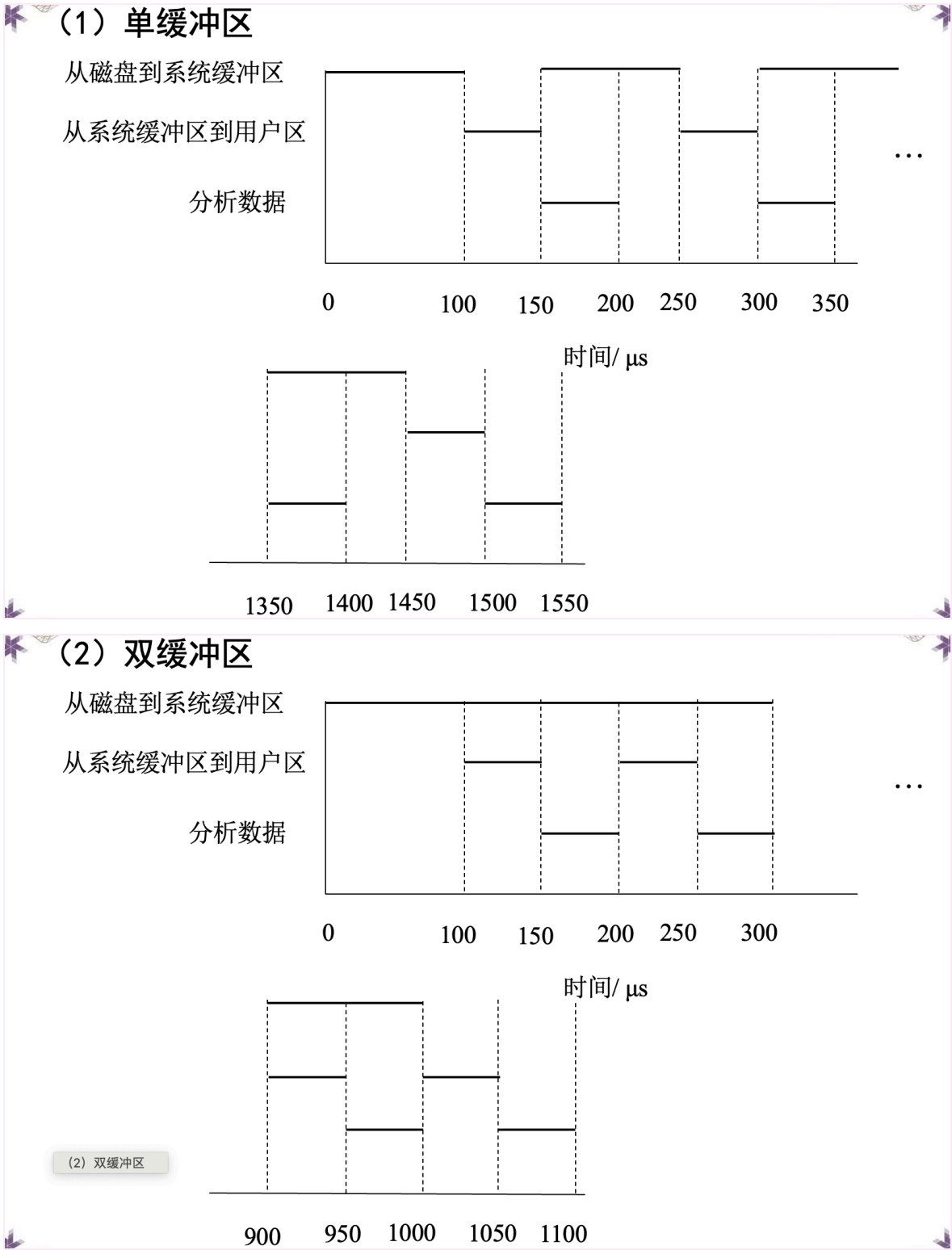


解析： 设备驱动程序接收与设备无关的软件发来的命令，并将命令中的抽象要求转换为与设备相关的低层操作序列。

17、磁盘访问时间包括包括**寻道时间、旋转延迟时间、传输时间。**

18、某文件占10个磁盘块，现要把该文件所占的磁盘块逐个地读入内存缓冲区，并送往用户区进行分析，假设一个缓冲区与一个磁盘块大小相同，把一个磁盘块读入缓冲区的时间为100μs，将缓冲区的数据传送到用户区的时间是50 μs，CPU对一块数据进行分析的时间为50 μs。在单缓冲区和双缓冲区结构下，读入并分析完该文件所需的时间分别是多少？

解析



19、先来先服务(FCFS): 有9个进程先后提出磁盘I/O请求, 磁头最先在100号磁道位置, 其I/O请求次序如下: 55, 58, 39, 18, 90, 160, 150, 38, 184。

解析: 先来先服务(FCFS): 55->58->39->18->90->160->150->38->184。

20、最短寻道时间优先(SSTF): 有9个进程先后提出磁盘I/O请求, 磁头最先在100号磁道位置, 其I/O请求次序如下: 55, 58, 39, 18, 90, 160, 150, 38, 184。

解析: 从小到大排列:

18 38 39 55 58 90 100 150 160 184

最短寻道时间优先(SSTF): 100->90->58->55->39->38->18->150->160->184

21、磁盘请求服务队列中要访问的磁道分别为38、6、37、100、14、124、65、67, 磁头上次访问了20磁道, 当前处于30磁道上, 试采用FCFS、SSTF和SCAN调度算法, 分别计算磁头移动的磁道数。

解析:

先排列顺序:

6 14 ~~20~~ 30 37 38 65 67 100 124

FCFS: 磁头移动的顺序为38、6、37、100、14、124、65、67, 移动磁道数为 $8 + 32 + 31 + 63 + 86 + 110 + 59 + 2 = 391$ 。

SSTF: 磁头移动的顺序为37、38、14、6、65、67、100、124, 移动磁道数为 $7 + 1 + 24 + 8 + 59 + 2 + 33 + 24 = 158$ 。

SCAN: 磁头移动的顺序为37、38、65、67、100、124、14、6, 移动磁道数为 $7 + 1 + 27 + 2 + 33 + 24 + 110 + 8 = 212$ 。

22、FCFS调度算法不会导致磁臂粘着

23、假设计算机系统采用CSCAN磁盘调度策略, 使用2KB的内存空间记录16384个磁盘块的空间状态。

(1) 请说明如何进行磁盘块空闲状态的管理。

(2) 设某单面磁盘的旋转速度为6000r/min。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移动, 磁道号请求队列为50、90、30、120。对请求队列中的每个磁道均须读取1个随机分布的扇区, 则读完这些扇区共需多少时间。

解析: 假设计算机系统采用CSCAN磁盘调度策略, 使用2KB的内存空间记录16384个磁盘块的空间状态。

(1) 因为 $2 \times 1024 \times 8 = 16384$ , 所以用每一位表示磁盘块是否空闲。

(2) 设某单面磁盘的旋转速度为6000r/min。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移动, 磁道号请求队列为50、90、30、120。对请求队列中的每个磁道均须读取1个随机分布的扇区。

每一转的时间为0.01s, 旋转延迟为0.005s, 通过一个扇区的时间(即传输时间)为0.0001s。被访问磁道号的顺序为100、120、30、50、90, 因此总寻道时间为 $(20 + 90 + 20 + 40) \times 1\text{ms} = 170\text{ms}$ 。总旋转延迟时间和总传输时间为 $(0.005 + 0.0001) \times 4 = 0.0204\text{s} = 20.4\text{ms}$ 。总时间为190.4ms。

## 第七章

1、逻辑文件是从用户观点出发的文件组织形式

- 2、数据库文件的逻辑结构形式是**记录式文件**
- 3、根据文件的逻辑组织可知，下列文件中**索引文件**是记录文件
- 4、**记录式文件**和**流式文件**可用于描述文件逻辑结构。
- 5、对于包含40000条记录的主数据文件，采用索引顺序文件组织方式，平均检索效率可**提高到**顺序文件组织方式的**多少倍？**

解： $\sqrt{40000 / 2} = 100$

如果在一个顺序文件中所含有的记录数为N，则为检索到具有指定关键字的记录，平均须查找N/2个记录。对于顺序索引文件，平均只要查找 $\sqrt{N}$ 个记录，因而其检索效率比顺序文件提高 $\sqrt{N/2}$ 倍。

6、已知yourfile文件的逻辑结构是由定长记录组成、按记录号排序的顺序文件，记录长度为128B。其中物理结构采用了顺序文件结构（即逻辑上连续的记录存放在连续的磁盘块中），文件的起始地址为200号磁盘块，磁盘块长512B。假设yourfile文件已经打开，要从打开的yourfile文件中读出18号记录（从0开始编号），请问：

- (1) 启动磁盘的次数是多少？每次读的磁盘块号是多少？
- (2) 块内位移量是多少？

解析：（1）：顺序文件是顺序存取文件，所以只需启动1次磁盘。

$512 / 128 = 4$ ，所以每个磁盘块可以存放4条记录。文件的起始地址是200号磁盘块，0-3，4-7，8-11，12-15，16-19这些记录分别在一个磁盘块上，所以18号记录在204号磁盘块上。

（2）：因为18号记录是16-19这个磁盘块的第三个记录，从0地址开始编制，16号记录占0-127，17号记录占128-255，所以18号记录的块内偏移量是256。

- 7、在文件系统中，文件访问控制信息所被存储的合理位置是**FCB**
- 8、在一个文件被用户首次打开的过程中，OS需要做的是**将FCB读到内存中**
- 9、使用绝对路径名访问文件是指从**根目录**开始按目录结构访问某个文件。
- 10、采用多级目录结构后，不同用户文件的文件名**相同或不同均可**
- 11、就文件的共享方式来说，**基于索引结点的共享方式**会在文件主删除其共享文件后留下悬空指针。
- 12、利用**符号链接**实现文件共享时，只有文件主才拥有其索引结点的指针，而共享同一文件的其它用户仅拥有对应文件的路径名，故不会在文件主删除其共享文件后留下悬空指针。
- 13、关于文件共享方式，不论是基于索引结点的共享方式还是基于符号链接的共享方式，均存在**遍历文件系统时多次遍历共享文件的问题**

## 第八章

- 1、按文件的物理结构可将文件分成**连续文件、链式文件、索引文件**
- 2、在文件系统中，若文件的物理结构采用连续结构，则FCB中有关文件的物理位置的信息包括**首块地址**和**文件长度**
- 3、若FAT16文件系统的簇和扇区大小分别为2KB和512B，则其所支持的磁盘分区容量为**128MB**

解： $2^{16} \times 2KB = 128MB$

4、某文件系统的簇和扇区大小分别为1KB和512B，若一个文件大小为1026B，则系统分配给该文件的磁盘空间的大小是**2048B**

解：分配两个簇

5、假定磁盘的盘块大小为1KB，若采用FAT文件系统进行管理，FAT表项大小要求为半字节的整数倍，且FAT占用的空间要求尽可能小，

- (1) 当磁盘分区容量为800MB时，其FAT占用多少存储空间？
- (2) 当磁盘分区容量为2GB时，其FAT占用多少存储空间？

解析：

(1)：总共有800M/1K = 800K个盘块， $2^{19} < 800K < 2^{20}$ 。因此，用20位表示盘块号，即每个FAT表项的大小为2.5B，因此，FAT占用的存储空间是800K × 2.5B = 2MB。

(2)：总共有2G/1K = 2<sup>21</sup>个盘块。因此，用24位表示盘块号，即每个FAT表项的大小为3B，因此，FAT占用的存储空间是2<sup>21</sup> × 3B = 6MB。

6、某FAT文件系统的簇大小为4KB。假定目录树如下：目录dir下有只有目录dir1，目录dir1下只有两个文件file1和file2。各文件占用的簇号及顺序如下：

文件名	簇号
dir	1
dir1	48
file1	100、106、108
file2	200、201、202

- (1) 请给出每个目录文件的目录项（只要给出文件名和簇号）
- (2) 若FAT的每个表项仅存放簇号，占2B，则FAT的最大长度为多少字节？该文件系统支持的文件长度最大是多少？
- (3) 系统通过目录文件和FAT实现对文件的按名存取，请说明file1的106、108两个簇号分别放在FAT的哪个表项中？
- (4) 假设FAT和dir目录文件已读入内存，若要将文件dir/dir1/file1的第5000个字节读入内存，则要访问哪几个簇？

解析：

- (1) 在FAT文件系统中，FCB中只存储第一个簇号，dir的目录项

文件名	簇号
dir1	48

- (2) 2B是16位，所以FAT的最大长度是2<sup>16</sup> × 2B = 128KB，文件的最大长度是2<sup>16</sup> × 4KB = 256MB。
- (3) 簇号106存放在100号表项中，簇号108存放在106号表项中。

(4)  $5000B = 4KB + 4B$ ，所以，该字节位于文件的第二个簇中，先访问dir1目录文件所在的48号簇，再通过FAT得到文件的第二个簇是106号簇，然后访问106号簇。

7、采用直接存取方法来读/写硬盘上的物理记录时，效率最低的文件是**链接文件**

解析：直接存取方法的访问效率：**顺序文件 > 索引文件 > 显式链接文件 > 隐式链接文件**

8、**索引结构**适合随机访问且易于文件扩展，**链接结构**不适合随机访问，**连续结构**不易于文件扩展。

9、为支持CD-ROM中视频文件的快速随机播放，播放性能最好的文件数据块组织方式是**连续结构**

10、设文件索引结点中有7个地址项，其中4个地址项为直接地址索引，两个地址项为一级间接地址索引，1个地址项为二级间接地址索引，每个盘块号和地址项都占4B，若索引块和盘块的大小都是256B，则可表示的单个文件的最大长度是**1057KB**

解析：一个索引块可以存放  $256 / 4 = 64$  个盘块号，直接地址项可以表示  $4 \times 256B = 1KB$ ，一级间址索引可以表示  $2 \times 64 \times 256B = 32KB$ ，二级间址索引可以表示  $64 \times 64 \times 256B = 1024KB$ ，总共1057KB。

11、在文件索引结点中有10个直接地址项，一级间接地址项和二级间接地址项各一个，每个盘块号和地址项都占4B，若索引块和盘块的大小都是1KB，则把该文件偏移量（按字节编址）为1234和307400处所在的盘块读入内存，访问磁盘的次数分别是**1、3**

**解析：**直接地址项可以表示10KB，一个索引块可以存储  $1KB / 4B = 256$  个盘块号，一级间接地址项可以表示  $256 \times 1KB = 256KB$ ，二级间接地址项可以表示  $256 \times 256 \times 1KB = 64MB$ ，

偏移量为1234时， $1234B < 10KB$ ，所以由直接地址项可得其所在盘块的地址，故访问一次磁盘。

偏移量为307400时， $307400B = 300KB + 200B$ ，所以  $10KB + 256KB < 307400B < 10KB + 256KB + 64MB$ ，所以该位置在二级间接地址项所指向的某个盘块中，故先访问2次磁盘得到该位置所在的盘块号，再访问1次磁盘读出内容，总共3次。

12、某文件系统采用混合索引方式组织文件的存储空间，盘块和索引块大小都是4KB，每个目录项中包括13个地址项，其中0-9是直接地址，10为一级间址，11为二级间址，12为三级间址。已知一个地址占4B，现有文件A、B、C，它们的大小分别为5KB、40.5KB、4138KB，若不计目录项，则请问系统分别给这些文件分配多大的磁盘空间？

**解析：**每个索引块可存储  $4KB / 4B = 1024$  个地址，

A文件：5KB需要两个盘块，所以需要两个直接地址项，分配8KB。

B文件：40.5KB需要11个盘块，所以需要10个直接地址项和1个一级间接地址索引块，分配12个盘块，48KB。

C文件：4138KB需要1035个盘块， $1035 - 10 = 1025$ ， $1025 - 1024 = 1$ ，所以需要10个直接地址项、1个一级间接地址索引块和两个二级间接地址索引块，总共1038个盘块，分配4152KB。

13、某文件系统采用索引结点存放文件的属性和地址信息，簇的大小是4KB，每个文件索引结点占64B，每个目录项中包括11个地址项，8个直接地址，一级间址、二级间址和三级间址各一个。已知一个地址占4B。

(1) 该文件系统能支持的最大文件长度是多少？（给出计算表达式即可）

(2) 文件系统采用1M个簇存放文件索引结点，用512M个簇存放文件数据。若一个图像文件的大小为5600B，则该文件系统最多能存放多少个这样的图像文件？

(3) 若文件F1的大小为6KB，文件F2的大小为40KB，则该文件系统获取F1和F2最后一个簇的簇号需要的时间是否相同？

**解析：**

(1) 每个簇可存放的地址项为  $4KB / 4B = 1024$  个。能支持的最大文件长度为  $(8 + 1024 + 1024^2 + 1024^3) \times 4KB = 32KB + 4MB + 4GB + 4TB$ 。

(2) 可以存储的索引结点数为  $1M \times 4KB / 64B = 64M$ 。5600B 占用两个簇，所以 512M 个簇可以存放文件 256M 个。

所以最多可以存储这样的文件 64M 个。

(3) F1 有 6KB，它的最后一个簇的簇号可以直接从直接地址项中得到，F2 有 40KB，要访问一级间接地址索引表才能得到最后一个簇的簇号，所以所需的时间不同。

14、现有容量为 10G 的磁盘分区，磁盘空间以簇为单位进行分配，簇的大小为 4KB，若采用位示图法管理该分区的空闲空间，则存放该位示图所需簇的个数为 **80**

**解析：**簇的总数为  $10GB / 4KB = 2.5M$ ，用一位标识一簇是否被分配，则整个磁盘共需要 2.5M 位，即需要  $2.5M / 8 = 320KB$ ，则共需要  $320KB / 4KB = 80$  个簇

15、空闲链表法可用于 **磁盘的空闲盘块组织**

16、文件系统采用位示图法表示磁盘空间的分配情况，位示图存放在磁盘的 32-127 号盘块中，每个盘块占 1024B，盘块和块内字节均从 0 开始编号。假设要释放的盘块号为 409612，则位示图中要修改的位所在盘块号和块内字节序号分别是 **82、1**

**解析：** $409612 / (8 \times 1024) = 50$ ， $409612 \% (8 \times 1024) = 12$ ，盘块号为  $32 + 50 = 82$ ，块内字节序号为  $12 / 8 = 1$ 。

17、关于文件存储空间的管理方式，**成组链接法要使用空闲盘块号栈。**