



第一章：

一、 3、 10、 15、 23、 27、 35

3.什么是操作系统？操作系统在计算机系统中的作用是什么？

操作系统是管理系统资源、控制程序执行、改善人机界面、提供各种服务,并合理组织计算机工作流程和为用户有效地使用计算机提供良好运行环境的一种系统软件。

主要作用

- (1) 服务用户 —操作系统作为用户接口和公共服务程序
- (2) 进程交互 —操作系统作为进程执行的控制者和协调者
- (3) 系统实现 —操作系统作为扩展机或虚拟机
- (4) 资源管理 —操作系统作为资源的管理者和控制者

10.试述系统调用与函数（过程）调用之间的区别。

- (1) 调用形式和实现方式不同；
- (2) 被调用的代码位置不同；
- (3) 提供方式不同

15.什么是多道程序设计？多道程序设计有什么特点？

多道程序设计是指允许多个作业（程序）同时进入计算机系统内存并执行交替计算的方法。从宏观上看是并行的，从微观上看是串行的。

- (1) 可以提高 CPU 内存和设备的利用率；
- (2) 可以提高系统的吞吐率，使单位时间内完成的作业数目增加；
- (3) 可以充分发挥系统的并行性，使设备和设备之间，设备和 CPU 之间均可并行工作。

23.现代操作系统具有哪些基本功能？请简单叙述之。

- (1) 处理器管理；
- (2) 存储管理；
- (3) 设备管理；
- (4) 文件管理；
- (5) 联网与通信管理。

27.什么是操作系统的内核？

内核是一组程序模块，作为可信软件来提供支持进程并发执行的基本功能和基本操作，通常驻留在内核空间，运行于内核态，具有直接访问计算机系统硬件设备和所有内存空间的权限，是仅有的能够执行特权指令的程序。

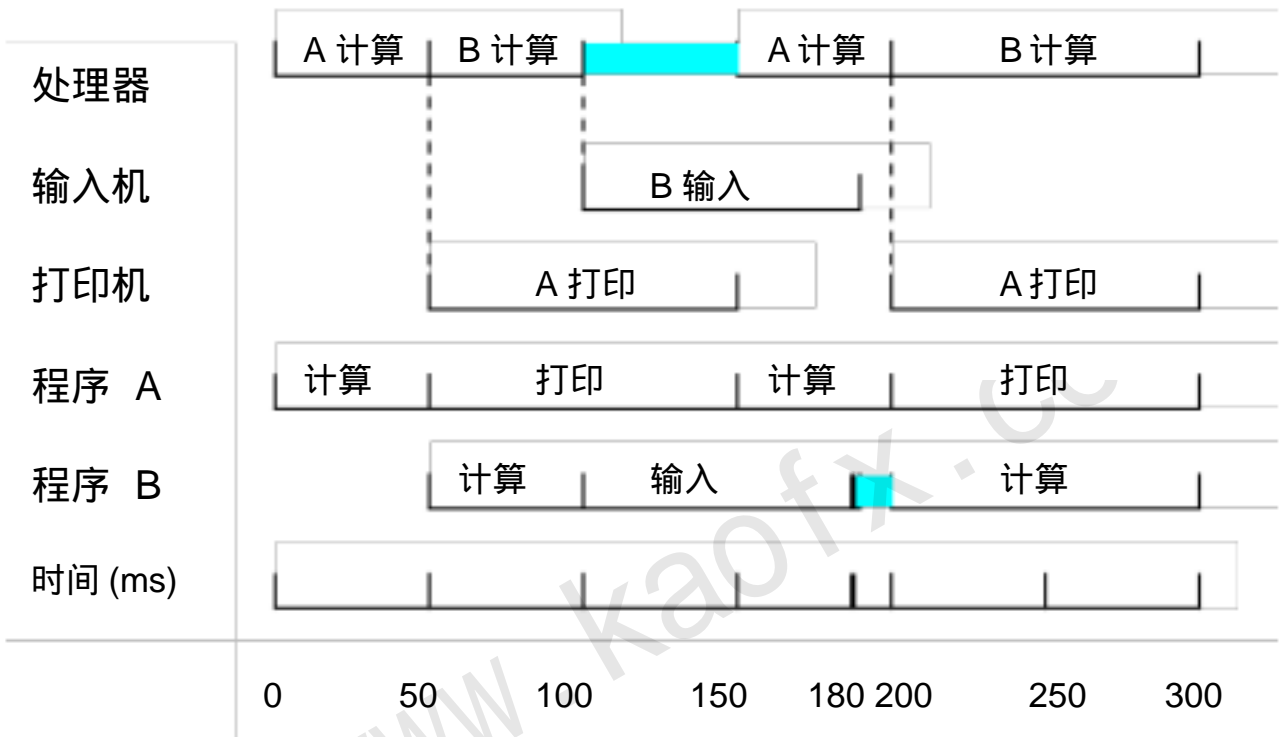
35.简述操作系统资源管理的资源复用技术。

系统中相应地有多个进程竞争使用资源，由于计算机系统的物理资源是宝贵和稀有的，操作系统让众多进程共享物理资源，这种共享称为资源复用。

- (1) 时分复用共享资源从时间上分割成更小的单位供进程使用；
- (2) 空分复用共享资源从空间上分割成更小的单位供进程使用。

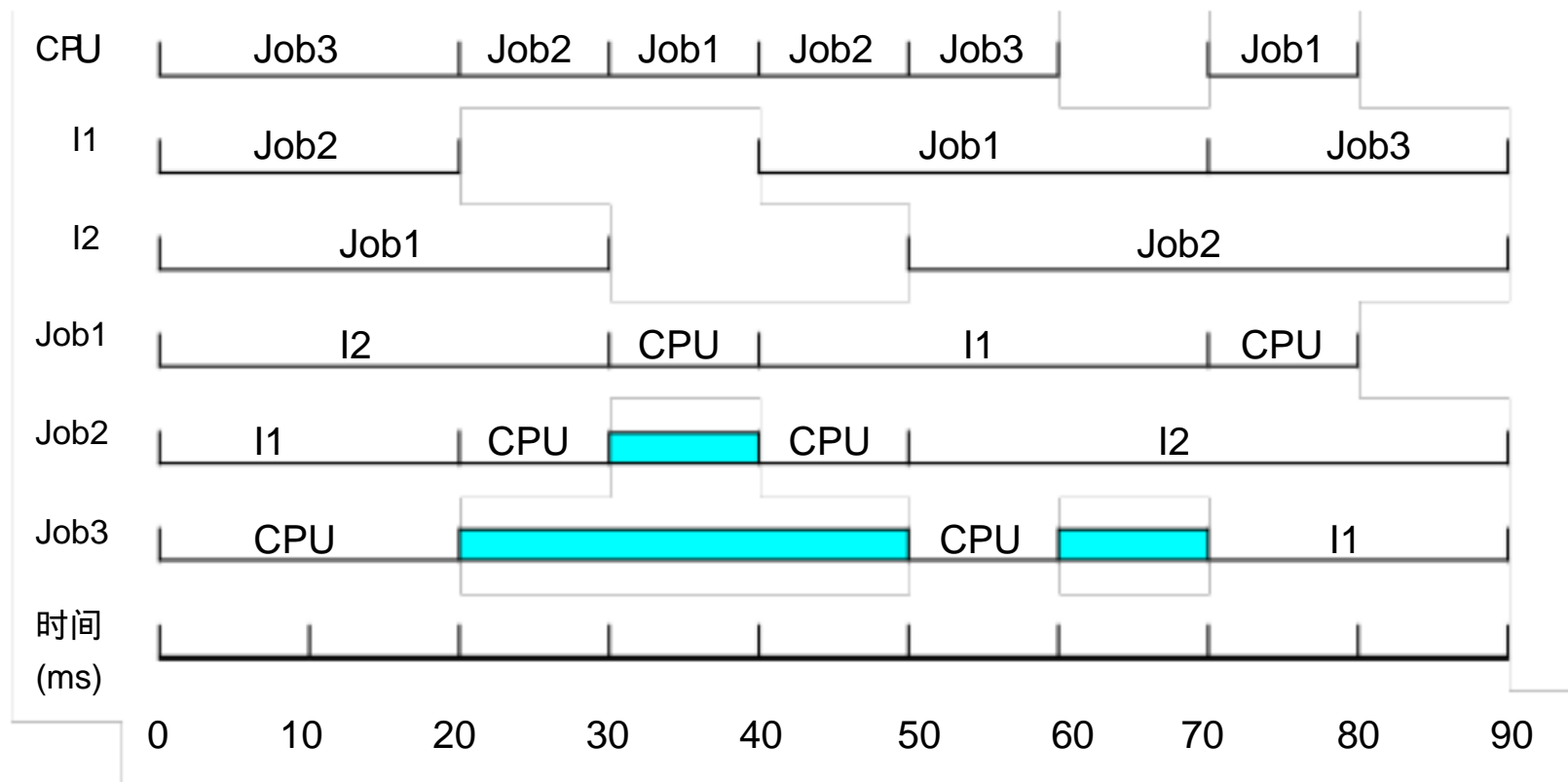
二、 2、 5

2、 答：画出两道程序并发执行图如下：



- (1) 两道程序运行期间，CPU 存在空闲等待，时间为 100 至 150ms 之间 (见图中有色部分)。
- (2) 程序 A 无等待现象，但程序 B 有等待。程序 B 有等待时间段为 180ms 至 200ms 间 (见图中有色部分)。

5、 答：画出三个作业并行工作图如下 (图中着色部分为作业等待时间)：



- (1) Job1从投入到运行完成需 80ms, Job2从投入到运行完成需 90ms, Job3从投入到运行完成需 90ms。
- (2) CPU 空闲时间段为：60ms 至 70ms, 80ms 至 90ms。所以 CPU 利用率为 $(90-20)/90=77.78\%$ 。
- (3) 设备 I1 空闲时间段为：20ms至 40ms, 故 I1 的利用率为 $(90-20)/90=77.78\%$ 。设备 I2 空闲时间段为：30ms至 50ms, 故 I2 的利用率为 $(90-20)/90=77.78\%$ 。

第二章：

一、9、18、20、26、48

9.什么是系统调用？试述 API、库函数和系统调用之间的关系。叙述系统调用执行流程。

18.什么是进程？计算机操作系统为什么要引入进程？

进程是具有独立功能的程序在某个数据集合上的一次运行活动，是计算机操作系统进行资源分配和保护的基本单位。

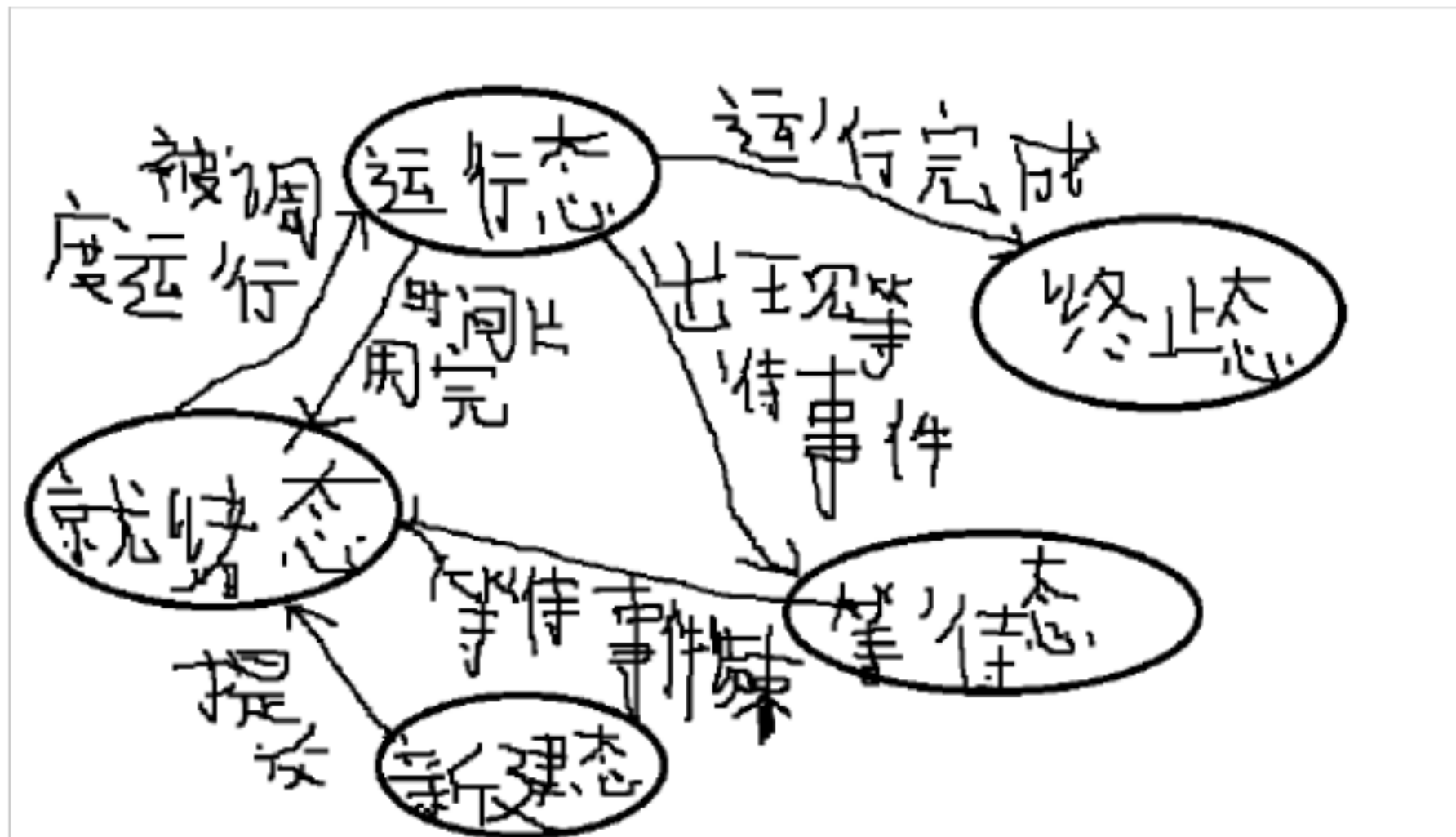
- (1) 刻画程序的并发性；
- (2) 解决资源的共享性。

20.进程的最基本状态有哪些？哪些事件可以引起不同状态间的转换？

运行态

就绪态

阻塞态



26. 何谓进程控制块 (PCB) ? 它包含哪些基本信息 ?

进程控制块是进程存在的唯一标识, 是操作系统用来记录和刻画进程状态及环境信息的数据结构, 是进程动态特征的汇集, 也是操作系统掌握进程的唯一资料结构和管理进程的主要依据。

- (1) 标识信息;
- (2) 现场信息;
- (3) 控制信息。

48. 处理器调度分为哪几种类型? 简述各类调度的主要任务。

高级调度, 中级调度, 低级调度。

高级调度, 创建就绪进程和做进程完成后的善后工作。

中级调度, 根据内存资源情况决定内存中所能容纳进程的数目, 并完成内存和外存中进程的对换。

低级调度, 根据某种原则决定就绪队列中哪个进程 / 线程获得处理器, 并将处理器出让给它使用。

二、 5、 10、 12、 16、 20、 21、 25、 28

5、 答: 采用短作业优先算法调度时, 三个作业的总周转时间为:

$$T1 = a + (a+b) + (a+b+c) = 3a + 2b + c$$

若不按短作业优先算法调度, 不失一般性, 设调度次序为: J2 J1、 J3 则三个作业的总周转时间为:



$$T2=b+(b+a)+(b+a+c)=3b+2a+c$$

令 - 式得到：

$$T2-T1=b-a>0$$

可见，采用短作业优先算法调度才能获得最小平均作业周转时间。

10、答：5 个待运行的作业同时到达，按照最短作业优先的算法可以使平均响应时间最短。X 取值不定，按照以下情况讨论：

- 1) $x < 3$ 次序为：x, 3, 5, 6, 9
- 2) $3 < x < 5$ 次序为：3, x, 5, 6, 9
- 3) $5 < x < 6$ 次序为：3, 5, x, 6, 9
- 4) $6 < x < 9$ 次序为：3, 5, 6, x, 9
- 5) $9 < x$ 次序为：3, 5, 6, 9, x

12、答：

(1)FCFS调度算法

执行次序	执行时间	等待时间	周转时间	带权周转时间
A	10	0	10	1
B	6	10	16	2.66
C	2	16	18	9
D	4	18	22	5.5
E	8	22	30	3.75
作业平均 周转时间			$T=(10+16+18+22+30)/5=19.2$	
作业平均 带权周转时间			$W=(1+2.66+9+5.5+3.75)/5=4.38$	

(3) 优先级调度算法

执行次序	执行时间	等待时间	周转时间	带权周转时间
B	6	0	6	1
E	8	6	14	1.75
A	10	14	24	2.4
C	2	24	26	13
D	4	26	30	7.5
作业平均 周转时间			$T=(6+14+24+26+30)/5=20$	
作业平均 带权周转时间			$W=(1+1.75+2.4+13+7.5)/5=5.13$	

(3)时间片轮转法 (每个作业获得相同的 2 分钟长的时间片)

按次序 A B C D E A B D E A B E A B C 轮转执行。



作业	执行时间	等待时间	周转时间	带权周转时间	
A		10	20	30	3
B		6	16	22	3.66
C		2	4	6	3
D		4	12	16	4
E		8	20	28	3.5
作业平均 周转时间				$T=(30+22+6+16+28)/5=20.4$	
作业平均 带权周转时间				$W=(3+3.66+3+4+3.5)/5=3.43$	

16、
答：

	FCFS			SJF			HRRF		
作业	开始时间	完成时间	周转时间	开始时间	完成时间	周转时间	开始时间	完成时间	周转时间
1	8:00	10:00	2.00	8:00	10:00	120	8:00	10:00	120
2	10:00	10:50	2.00	10:30	11:20	150	10:10	11:00	130
3	10:50	11:00	2.00	10:00	10:10	70	10:00	10:10	70
4	11:00	11:20	1.5	10:10	10:30	40	11:00	11:20	90
平均周转时间 =	T=112.5 分			T=95 分			T=102.5 分		
带权平均周转时间 =	W=4.975			W=3.25			W=3.775		

20、答：

执行次序	提交时间	执行时间	开始时间	完成时间	周转时间	
J1	8:00	60	8:00	9:00	60	
J5	8:35	5	9:00	9:05	30	
J6	8:40	10	9:05	9:15	35	
J3	8:25	20	9:15	9:35	70	
J4	8:30	25	9:35	10:00	90	
J2	8:20	35	10:00	10:35	135	
作业平均 周转时间 $T=(60+30+35+70+90+135)/6=70$						

注意，J1 被调度运行后，直到它执行结束，才会引出作业调度程序工作。所以，J2 至 J6 虽在 J1 执行期间进入，但未被调度，均在等待。当 J1 撤离后，作业调度程序工作，按 SJF 算法，显然有执行次序：J5、J6、J3、J4、和 J2。

21

作业名	到达时刻	估计运行时间 /min	优先数
JOB1	10:00	40	5
JOB2	10:20	30	3
JOB3	10:30	60	4
JOB4	10:50	20	6
JOB5	11:00	20	4
JOB6	11:10	10	4

答：如下表所示

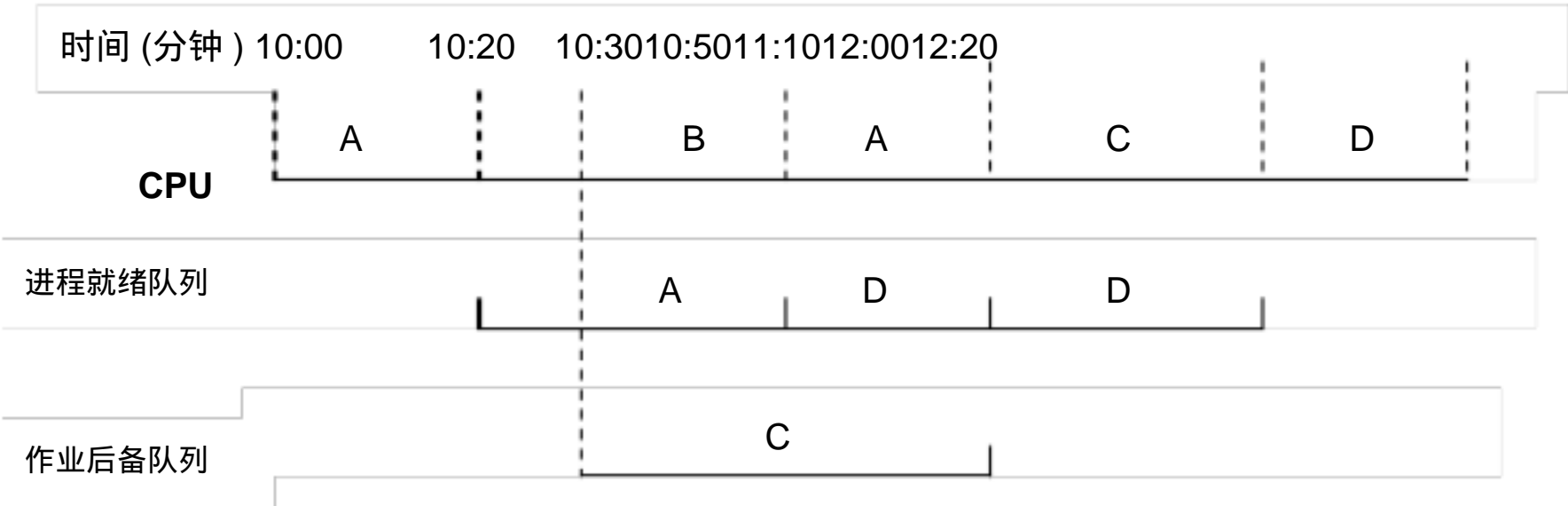
作业名	进入内存时刻	开始运行时刻	结束运行时刻	周转时间	带权周转时间
JOB1	10:00	10:00	12:40	160	4
JOB2	10:20	10:20	10:50	30	1
JOB3	10:30	10:50	11:50	80	4/3
JOB4	10:50	12:40	13:00	130	13/2
JOB5	12:00	12:00	12:20	80	4
JOB6	11 : 50	11:50	12:00	50	5

平均周转时间 =(160+30+80+130+80+50)/6=88.88

平均带权周转时间 =(4+1+4/3+13/2+4+5)/6=3.64

25、答：

每个作业运行将经过两个阶段：作业调度 (SJF 算法)和进程调度 (优先数抢占式)。另外，批处理最多容纳 2 道作业，更多的作业将在后备队列等待。





- (1) 10:00, 作业 A 到达并投入运行。
- (2) 10:20, 作业 B 到达且优先权高于作业 A, 故作业 B 投入运行而作业 A 在就绪队列等待。
- (3) 10:30, 作业 C 到达, 因内存中已有两道作业, 故作业 C 进入作业后备队列等待。
- (4) 10:50, 作业 B 运行结束, 作业 D 到达, 按 SJF 短作业优先算法, 作业 D 被装入内存进入就绪队列。而由于作业 A 的优先级高于作业 D, 故作业 A 投入运行。
- (5) 11:10, 作业 A 运行结束, 作业 C 被调入内存, 且作业 C 的优先级高于作业 D, 故作业 C 投入运行。
- (6) 12:00, 作业 C 运行结束, 作业 D 投入运行。
- (7) 12:20, 作业 D 运行结束。

作业进入	内存时间运行结束时间	
A	10:0011:10	10:50
B	10:20	
C	11:1012:00	
D	10:5012:20	

各作业周转时间为：作业 A 70, 作业 B 30, 作业 C 90, 作业 D 90。平均作业周转时间为 70 分钟。

28、答：

- (1) FIFO 算法选中作业执行的次序为：A、B、D、C 和 E。作业平均周转时间为 63 分钟。
- (2) SJF 算法选中作业执行的次序为：A、B、D、E 和 C。作业平均周转时间为 58 分钟。

第三章：

一、9、13、15、25

9.什么是临界区和临界资源？临界区管理的基本原则是什么？

并发进程中与共享变量有关的程序段称为临界区。共享变量所代表的资源叫做临界资源，即一次仅供一个进程使用的资源。

- (1) 一次至多有一个进程进入临界区内执行；
- (2) 如果已有进程在临界区内，试图进入此临界区的其它进程应等待；
- (3) 进入临界区的进程应在有限时间内退出，以便让进程等待队列中的一个进程进入。

13.什么是信号量？如何对其进行分类？

信号量是物理资源的实体，它是一个与队列有关的整型变量。

按用途分



- (1) 公用信号量；
- (2) 私有信号量。

按取值分

- (1) 二值信号量；
- (2) 一般信号量。

15.何谓管程？它有哪些属性？

管程是由局部于自己的若干公共变量及其声明和所有访问这些公共变量的过程所组成的软件模块，它提供一种互斥机制，进程可以互斥地调用管程的过程。

- (1) 共享性；
- (2) 安全性；
- (3) 互斥性。

25.试述产生死锁的必要条件、死锁产生的原因及预防死锁的方法。

- (1) 互斥条件；
- (2) 占有和等待条件；
- (3) 不剥夺条件；
- (4) 循环等待条件。

进程推进顺序不当、PV 操作使用不妥、同类资源分配不均或对某些资源的使用未加限制等，不仅与系统拥有的资源数量有关，而且与资源分配策略、进程对资源的使用要求以及并发进程的推进顺序有关。

- (1) 破坏条件 1 (互斥条件)；
- (2) 破坏条件 2 (占有和等待条件)；
- (3) 破坏条件 3 (不剥夺条件)；
- (4) 破坏条件 4 (循环等待条件)。

二、

2、

答：不同

- (1) :初值为 1，范围为 $[-n+1,1]$ ；(2) :初值为 m ，范围为 $[-n+m,m]$ 。

5、答：1)使用信号量和 P、V 操作：

```
var  name: array[1..100] of A;
    A=record
        number:integer;
        name:string;
    end
    for i:=1 to 100 do {A[i].number:=i; A[i].name:=null;}
```



```

    mutex,seatcount:semaphore;
    i:integer;mutex:=1;seatcount:=100;
cobegin
{
    process  readeri(varreadername:string)(i=1,2, , )
    {
        P(seatcount);
        P(mutex);
        for i:=1 to 100 do i++
        if A[i].name=null then A[i].name:=readername;
        reader get the seat number =i;          /*A[i].number
        V(mutex)
        进入阅览室，座位号    i，座下读书 ；
        P(mutex);
        A[i] name:=null;
        V(mutex);
        V(seatcount);
        离开阅览室 ；
    }
}
coend.

```

2) 使用管程操作：

TYPE readbook=monitor

VAR R:condition;

Interface Module IM;

i,seatcount:integer;

name:array[1..100] of string;

DEFINE readercome,readerleave;

USE check,wait,signal,release;

procedure readercome(readername)

begin

check(IM);

if seatcount = 100 wait(R,IM)

seatcount:=seatcount+1;

for i=1 to 100 do i++

if name[i]==null then name[i]:=readername;

get the seat number=i;

release(IM);

end

procedure readerleave(readername)

begin



```

    check(IM);
seatcount--;
    for i=1 to 100 do i++
        if name[i]==readername then name[i]:=null;
        release(IM);
    end
end

```

```

begin
seatcount:=100;name:=null;
end

```

```

cobegin
    process readeri(i=1,2.    ... )
        begin
readercome(readername);
            read the book;
readerleave(readername);
            leave the readroom;
        end
    coend.

```

6、答：实质上是两个进程的同步问题，设信号量 **S1** 和 **S2** 分别表示可拣白子和黑子，不失一般性，若令先拣白子。

```

var  S1,S2:semaphore;
    S1:=1;S2:=0;

```

```

cobegin
{
    process P1
        begin
            repeat
                P(S1);
                拣白子
                V(S2);
            until false;
        end
    process P2
        begin
            repeat
                P(S2);
                拣黑子
                V(S1);
            until false;
        end
    }

```



coend.

16、答：(1)用信号量和 P、V 操作。

var S,S1,S2,S3;semaphore;

S:=1;S1:=S2:=S3:=0;

flag1,flag2,flag3:Boolean;

flag1:=flag2:=flag3:=true;

cobegin

{

process 供应者

begin

repeat

P(S);

取两样香烟原料放桌上，由 flagi 标记；/*flage1、flage2、flage3 代表烟草、纸、火柴

if flag2&flag3 then V(S1); /* 供纸和火柴

else if flag1&flag3 then V(S2); /* 供烟草和火柴

else V(S3); /* 供烟草和纸

until false;

end

process 吸烟者 1

begin

repeat

P(S1);

取原料；

做香烟；

V(S);

吸香烟；

until false;

process 吸烟者 2

begin

repeat

P(S2);

取原料；

做香烟；

V(S);

吸香烟；

until false;

process 吸烟者 3

begin

repeat

P(S3);

取原料；

做香烟；

V(S);



```

吸烟 ;
until false;
}
coend.

```

23、

答：(1) P1, P2, P3, P4 的 Cki-Aki 分别为：(2, 2, 2)、(1, 0, 2)、(1, 0, 3)、(4, 2, 0)

(1) 系统处于安全状态，存在安全序：P2, P1, P3, P4

(2) 可以分配，存在安全序列：P2, P1, P3, P4。

(3) 不可以分配。

24、

答：(1)系统处于安全状态，存在安全序列：P0, P3, P4, P1, P2。

(2)不能分配，否则系统会处于不安全状态。

25、

答：(1)此时可以找出进程安全序列：P4, P1, P5, P2, P3。故系统处于安全状态。

(2)可以分配，存在安全序列：P4, P1, P5, P2, P3。

(3)不可分配，系统进入不安全状态。

31、答：

```
semaphore wait, mutex1, mutex2;
```

```
mutex1=mutex2=1; wait=1;
```

```
int counter1, counter2;
```

```
count1=count2=0;
```

```
Cobegin
```

```
{
```

```
Process p1
```

```
{
```

```
P(mutex1);
```

```
counter1++;
```

```
if (counter1==1) then P(wait);
```

```
V(mutex1);
```

```
过独木桥;
```

```
P(mutex1);
```

```
counter1--;
```

```
if (counter1==0) then V(wait);
```

```
V(mutex1);
```




```

    }
    Process p2
    {
        P(mutex2) ;
        Counter2++ ;
        if (counter2==1) then P(wait) ;
        V(mutex2) ;
        过独木桥 ;
        P(mutex2) ;
        counter2-- ;
        if (counter2==0) then V(wait) ;
        V(mutex2) ;
    }
}
coend

```

50、答：

答：semaphore mutex1 , mutex2 ;

semaphore empty , full , count ;

mutex1=mutex2=1 ; count=3 ; empty=10 ; full=0 ;

cobegin

process youngmonki(i=1, 2..)

{

While(true){

 P(empty) ;

 P(count) ;

 P(mutex1) ;

 从井中取水 ;

 V(mutex1) ;

 P(mutex2) ;

 倒入水缸 ;

 V(mutex2) ;

 V(count) ;

 V(full) ;

 }

}

coend

process oldmonki(i=1, 2..)

{

while(true){

 P(full) ;

 P(count) ;

 P(mutex2) ;

 从缸中取水 ;

 V(mutex2) ;

 V(count) ;

 V(empty) ;

 }

}



第四章：

一、

1、试述存储管理的基本功能。

答：

- (1) 存储分配；
- (2) 地址映射；
- (3) 存储保护；
- (4) 存储共享；
- (5) 存储扩充。

4、何谓地址转换（重定位）？哪些方法可以实现地址转换？

答：

逻辑地址转换为物理地址的过程称为地址转换（重定位）。

- (1) 静态地址重定位；
- (2) 动态地址重定位；
- (3) 运行时链接地址重定位。

18、试述实现虚拟存储器的基本原理。有哪几种实现方法？

答：

虚拟存储器是指在具有层次结构存储器的计算机系统中，自动实现部分装入和部分替换功能，能从逻辑上为用户提供一个比物理内存容量大得多的、可寻址的“内存存储器”。是一种具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。

虚拟存储器的实现方式有两种：请求分页系统和请求分段系统。请求分页系统允许只装入少数页面的程序（及数据），便启动运行，以后，再通过调页功能及页面置换功能，陆续地把即将要运行的页面调入内存，同时把暂不运行的页面换出到外存上；请求分段系统允许只装入少数段（而非所有的段）的用户程序和数据，即可启动运行。以后再通过调段功能和段的置换功能将暂不运行的段调出，同时调入即将运行的段。

二、

1、答：

- (1) 当有 3 个物理页面时：

使用 FIFO 算法：缺页 16 次，缺页中断率 $16/20=80\%$ ；

使用 OPT 算法：缺页 11 次，缺页率 $11/20=55\%$ 。

使用 LRU 算法：缺页 12 次，缺页率 $12/20=60\%$ 。

- (2) 当使用 4 个物理页面时：

- (2) 当使用 5 个物理页面时：

- (2) 当使用 6 个物理页面时：



3、答：(1) 作业的物理块数为 3 块，使用 FIFO 为 9 次， $9/12=75\%$ 。使用 LRU 为 7 次， $7/12=58\%$ 。使用 OPT 为 6 次， $6/12=50\%$ 。

作业的物理块数为 4 块，使用 FIFO 为 6 次， $6/12=50\%$ 。使用 LRU 为 6 次， $6/12=50\%$ 。使用 OPT 为 5 次， $5/12=42\%$ 。

(2) 作业的物理块数为 3 块，使用 FIFO 为 9 次， $9/12=75\%$ 。使用 LRU 为 10 次， $10/12=83\%$ 。使用 OPT 为 7 次， $7/12=58\%$ 。

作业的物理块数为 4 块，使用 FIFO 为 10 次， $10/12=83\%$ 。使用 LRU 为 8 次， $8/12=66\%$ 。使用 OPT 为 6 次， $6/12=50\%$ 。

其中，出现了 Belady 现象，增加分给作业的内存块数，反使缺页中断率上升。

29、

答：1)680 2)915 3)904 4)越界 5)1750 6)越界。

30、

答：1)1) 页面访问序列为 0, 0, 1, 1, 0, 3, 1, 2, 2, 4, 4, 3。

2)FIFO 为 5 次，缺页中断率为 $5/12=41.6\%$ 。LRU 为 6 次，缺页中断率为 $6/12=50\%$ 。LRU 反比 FIFO 缺页中断率高。

第五章：

一、

1、试述设备管理的基本功能。

答：P255

- (1) 设备中断处理；
- (2) 缓冲区管理；
- (3) 设备分配与去分配；
- (4) 设备驱动调度；
- (5) 实现虚拟设备。

12、为什么要引入缓冲技术？其基本思想是什么？

答 P265

为了解决 CPU 与设备之间速度不匹配的矛盾及逻辑记录和物理记录大小不一致的问题，提高 CPU 与设备的并行性，减少 I/O 操作对 CPU 的中断次数，放宽对 CPU 中断响应时间的要求，人们为操作系统引入了缓冲技术。

当进程执行写操作输出数据时，先向系统申请一个输出缓冲区，然后将数据送至缓冲区，如果是顺序写请求，则不断地把数据填入缓冲区，直至装满为止，此后进程可以继续计算，同时系统将缓冲区数据写入设备。当进程执行读操作输入数据时，先向系统申请一个输入缓冲区，然后系统将设备的一条物理记录读至缓冲区，期间进程继续计算，送至完成之后系统根据要求将当前进程所需要的逻辑记录从缓冲区中选出并传送给进程。

21、什么是虚拟设备？实现虚拟设备的主要条件是什么？



答：

虚拟设备：为了提高独占设备的利用率，采用 **SPOOLING** 技术，用可共享的设备模拟独占设备，使独占设备成为共享设备，使每个作业感到自己分到了速度极高的独占设备。这种模拟的独占设备称为虚拟设备。

- (1) 预输入程序；
- (2) 井管理程序；
- (3) 缓输出程序。

28、为什么要引入设备独立性？如何实现设备独立性？

答：

设备独立性：用户不指定物理设备，而是指定逻辑设备，使得用户作业和物理设备之间分离开来，再通过其他途径建立逻辑设备和物理设备之间的映射，设备的这种特性就是“设备无关性”。

应用程序与具体物理设备无关，系统增减或变更设备时对源程序不必加以修改；易于应对 I/O 设备故障，提高系统可靠性；增加设备分配的灵活性，更有效地利用逻辑设备资源，实现多道程序设计。

为了实现设备独立性，必须在设备驱动程序之上设置一层设备独立性软件，用来执行所有 I/O 设备的公用操作，并向用户层软件提供统一接口。关键是系统中必须设置一张逻辑设备表 **LUT** 用来进行逻辑设备到物理设备的映射，其中每个表目中包含了逻辑设备名、物理设备名和设备驱动程序入口地址三项；当应用程序用逻辑设备名请求分配 I/O 设备时，系统必须为它分配相应的物理设备，并在 **LUT** 中建立一个表目，以后进程利用该逻辑设备名请求 I/O 操作时，便可从 **LUT** 中得到物理设备名和驱动程序入口地址。

30、目前常用的磁盘驱动调度算法有哪几种？分别适用于何种数据应用场合？

答：只需回答几处算法。

- (1) 电梯 I/O 调度算法；
- (2) 时限 i/o 调度和预期 i/o 调度算法；
- (3) 公平排队 i/o 调度算法。

- (1) 先来先服务调度算法；
- (2) 最短寻道时间调度算法；
- (3) 扫描算法；
- (4) 循环扫描算法。

二、

2、

答：处理次序为：100-110-129-147-186-78-64-41-27-18-12-10-8。移动的总柱面数：264。

7、

答：(1) 先来先服务算法 **FCFS** 为 565，依次为 143-86-147-91-177-94-150-102-175-130。

(2) 最短查找时间优先算法 **SSTF** 为 162，依次为 143-147-150-130-102-94-91-86-175-177。



(3)扫描算法 **SCAN**为 169 , 依次为 143-147-150-175-177-199-130-102-94-91-86 。

(4)电梯调度为 125(先向地址大的方向) ,依次为 143-147-150-175-177-102-94-91-86 。
为 148(先向地址小的方向) 依次为 143-130-102-94-91-86-147-150-175-177 。

10

答：采用先来先服务 处理次序为： 100-23-376-205-132-19-61-190-398-29-4-18-40 ，总柱面数为： 1596。

采用 **SSTF**处理次序为： 100-132-190-205-61-40-29-23-19-18-4-376-398 ，总柱面数为： 700。

采用 **SCAN**处理次序为： 100-132-190-205-376-398-61-40-29-23-19-18-4 ，总柱面数为： 692。

第六章：

一、 4、什么是文件的逻辑结构？它有哪些组织方式？

文件的数据独立于物理环境构造，这样的文件结构称为文件的逻辑结构。

(1) 流式文件；

(2) 记录式文件

A . 记录式顺序文件

B . 记录式索引顺序文件

5、什么是文件的物理结构？它有哪些组织方式？

答：

逻辑文件在物理存储空间中映射的文件结构称为文件的物理结构。

组织方式

(1) 顺序文件

(2) 连接文件

(3) 直接文件

(4) 索引文件

16、什么是文件共享？介绍文件共享的分类和实现方法。

答：文件共享是指不同进程共同使用同一个文件。

文件共享分类及实现方法：

静态共享：操作系统允许一个文件同时属于多个目录，但实际上文件仅有一处物理存储

动态共享：系统中不同的应用进程或同一用户的不同进程并发的访问通一个文件，共享关系在进程存在时出现。进程消亡时消亡

符号链接共享：符号链接只有文件名，不指向 **inode** 链接，通过名称来引用文件。

二、

3、

18



答：(1) 位示图占用字数为 $500/32=16$ (向上取整)个字。

(2) 第 i 字第 j 位对应的块号 $N=32 \times i+j$ 。

(3) 申请时自上至下、自左至右扫描位示图跳过为 1 的位，找到第一个迁到的 0 位，根据它是第 i 字第 j 位算出对应块号，并分配出去。归还时已知块号，块号 $/32$ 算出第 i 字第 j 位并把位示图相应位清 0。

8、

答： $1569/512$ 得到商为：3，余数为：33。所以，访问的是 80 磁盘块的第 33 个字节。

16、

由于索引节点为 128B，而状态信息占用 68B，故索引节点中用于磁盘指针的空间大小为：

$128-68=60$ 字节。

一次间接、二次间接和三次间接指针占用三个指针项，因而直接指针项数为： $60/4-3=12$ 个。每块大小为 8KB。所以，直接指针时： $12 \times 8192=98304B$ 。

一次间接指针时： $8192/4=2048$ ，即一个磁盘块可装 2048 个盘块指针， $2048 \times 8192=16MB$ 。

二次间接指针时： $2048 \times 2048=4M$ ，即二次间接可装 4M 个盘块指针， $4M \times 8192=32GB$ 。

三次间接指针时： $2048 \times 2048 \times 2048=8G$ ，即三次间接可装 8G 个盘块指针， $8G \times 8192=16TB$ 。