

补充习题

- 一、作业的周转时间
- 二、计算作业周转时间
- 三、P/V操作例子
- 四、银行家算法
- 五、存储分配与地址变换
- 六、磁盘调度及设备管理概念
- 七、磁盘空闲块的成组分配算法
- 八、概念复习题

练习题

- 一个32位的虚拟存储系统有两级页表，其逻辑地址中，第22到31位是第一级页表，12位到21位是第二级页表，页内偏移占0到11位。一个进程的地址空间为4GB，如果从0XC0000000开始映射4MB大小页表，请问第一级页表所占的4KB空间映射在什么位置，并说明理由。（注意B代表字节，一个32位地址占4字节）



二、作业的周转时间

有5个作业，到达的时间、所需执行时间、所需存储容量如下表所示。假定**内存容量为100KB**，采用可变分区存储管理方法，作业在内存中不能移动，并且假定内存中的**就绪作业平分CPU时间**，采用**先来先服务**调度策略，试求各作业开始执行的时间、完成的时间和周转时间，并填入表中。（不计作业换进换出时间和其它的开销）

作业号	作业提交时间	所需CPU时间(分)	作业长度(KB)	开始执行时间	作业完成时间	周转时间(分) T=完成-提交
1	10: 00	25	15	10:00	10:25	25
2	10: 20	30	60	10:25	11:20	60
3	10: 20	25	40	10:50	11:15	55
4	10: 30	15	20	11:20	11:45	75
5	10: 35	10	30	11:30	11:40	65

$$\text{平均周转时间} = (25+60+55+75+65) / 5 = 56 \text{ (分钟)}$$

二、计算作业周转时间

操作系统中采用多道程序设计技术以提高CPU及各种资源的利用率, 试以下例比较说明.

假定系统中有一个CPU, 一台I/O设备, 二者可同时工作, 有三个作业X,Y,Z, 各自的程序行为是:

	I/O	CPU	I/O	CPU	I/O	(单位: ms)
X	20	20	20	30	20	----->
Y		40	30	40	10	----->
Z		30	20	30		----->

- ① 假若采用单道串行方式调度运行, 总的周转时间是多少?
- ② 假若采用多道程序设计技术调度, 调度优先权X最高, Y其次, Z最低, 当前三个作业都已就绪. 试画出调度时序图, 给出各自的以及总的周转时间.



二(续)

①各自的周转时间

$$X = (20+20+20+30+20) = 110 \text{ (ms)}$$

$$Y = (20+20+20+30+20) + (40+30+40+10) = 230 \text{ (ms)}$$

$$Z = (20+20+20+30+20) + (40+30+40+10) + (30+20+30) = 310 \text{ (ms)}$$

总的周转时间=110+230+310=570, 平均周转时间=570/3=190

②各自的周转时间

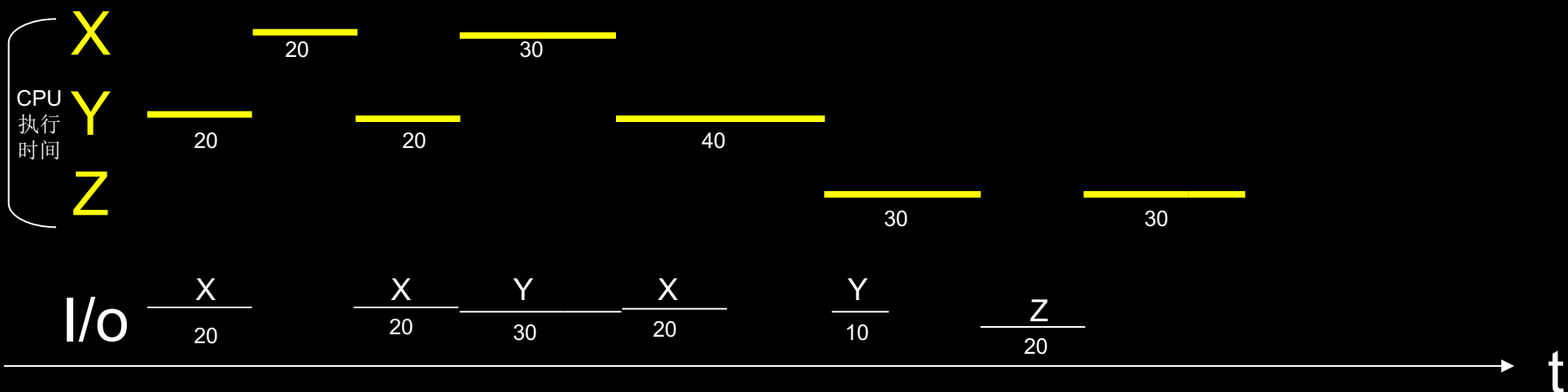
X:110(ms)

Y:140 (ms)

Z:210(ms)

总的=110+140+210=460, 平均=460/3≈153

	I/O	CPU	I/O	CPU	I/O (单位: ms)
X	20	20	20	30	20
Y		40	30	40	10
Z		30	20	30	



三、P/V操作例子

(司机与售票员)

汽车司机与售票员之间必须协同工作,一方面,只有售票员把车门关好了,司机才能开车,因此,售票员关好车门应通知司机开车;另一方面,只有当汽车已经停下,售票员才能开门上下客,故司机停车后应通知售票员.假定某辆公共汽车上有一名司机与两名售票员.汽车当前正在始发站停车上客.试设必要的信号量并赋初值,用P、V操作写出他们的同步算法.

三 (续)

设信号量为: **doorclose=0; busstop=1;**

conbegin

Driver:begin

repeat

P(doorclose);

开车;

到站;

停车;

V(busstop);

乘客上/下车;

forever

end

conductor:begin

repeat

P(busstop);

开门;

乘客上/下车;

售票;

关门;

V(doorclose);

开车;

forever

end

conend



习题：一个供销商与三个喝冰水者

- 口渴的人必须有三样东西：水、冰和茶杯才能喝到冰水。
- 有三个人，每人手中仅有一样上述东西
- 第四个人是服务员
 - 该人可以无限地提供这三样东西
 - 没有人喝水时，服务员便随机地提供其中的两样东西放在桌上
 - 如果这两样东西是口渴的人所需要的，则按需要收起，便可喝一杯冰水。喝完后则通知服务员，此过程反复进行。
- 写一个管程，控制口渴者和服务员的活动过程。



习题解答：一个供销商与三个喝冰水者

```
monitor drinkers()
{
    condition iceWater, iceGlass, waterGlass;
    condition server();
    boolean suppliedIce = false;
    boolean suppliedWater = false;
    boolean suppliedGlass = false;
    typedef requirements (water, ice, glass);
    int drinkers = 0;
    boolean first = true;

    Serve()
    {
        if (first == false)
            server._wait();
        first = false;
        rr = random(3);
        if (rr == 0)
        {
            suppliedIce = true;
            suppliedWater = true;
            iceWater._signal();
        }
        else if (rr == 1)
        {
            suppliedGlass = true;
            suppliedIce = true;

            iceGlass._signal();
        }
        else
        {
            suppliedGlass = true;
            suppliedWater = true;
            waterGlass._signal();
        }
    }

    GetIngredients(requirements has)
    {
        if (has == glass)
        {
            if (suppliedIce == false || suppliedWater == false)
                iceWater._wait();
        }
        else if (has == water)
        {
            if (suppliedIce == false || suppliedGlass == false)
                iceGlass._wait();
        }
        else
        {
            if (suppliedGlass == false || suppliedWater == false)
                waterGlass._wait();
        }
        suppliedIce = false;
        suppliedWater = false;
        suppliedGlass = false;
    }

    NotifyServer()
    {
        server._signal();
    }
}
```

Procedure drinker1

While(true)

```
{  
    drinkers.GetIngredients(requirements water);  
    drink();  
    drinkers.NotifyServer();  
}
```

Procedure drinker2

While(true)

```
{  
    drinkers.GetIngredients(requirements ice) ;  
    drink();  
    drinkers.NotifyServer();  
}
```

Procedure drinker3

While(true)

```
{  
    drinkers.GetIngredients(requirements glass) ;  
    drink();  
    drinkers.NotifyServer();  
}
```

Procedure sever

While(true)

```
{  
    drinkers. Serve();  
}
```

Cobegin

```
    sever;  
    drinker1;  
    drinker2;  
    drinker3;
```

coend

P/V操作的应用

- 某公司有两个生产部门和一个装配部门，两个生产部门分别生产甲、乙两种零件，装配部门的任务是把甲、乙两种零件组装成产品。两个生产部门每生产一个零件后都要分别把它们送到装配部门的货架S1、S2上。S1存放零件甲，S2存放零件乙，S1和S2均可容纳20个零件。装配人员每次从货架上取一个甲零件和一个乙零件后组装成产品。请用P、V操作进行正确管理。



算法描述

```
Begin
  信号量初值:
  mutex1:=1;
  mutex2:=1;
  empty1:=20;
  empty2:=20;
  full1:=0;
  full2:=0

  Cobegin
    A部门:
    begin
      Repeat
        生产一个产品A;
        P(empty1);
        P(mutex1);
        将产品A放入S1;
        V(mutex1);
        V(full1);
      Until false
    End
```

```
    B部门:
    begin
      Repeat
        生产一个产品B;
        P(empty2);
        P(mutex2);
        将产品B放入S2;
        V(mutex2);
        V(full2);
      Until false
    End
```

```
  装配人员: begin
    Repeat
      P(full1);
      P(full2);
      P(mutex1);
      从S1中取出产品A;
      V(mutex1);
      V(empty1);
      P(mutex2);
      从S2中取出产品B;
      V(mutex2);
      V(empty2);
      把A和B组装成产品
    Until false
  End
Coend
End;
```



另一种描述?

Begin

A:=1;

B:=1;

S1:=0;

S2:=0;

Procedure:

Product A: 部门A

begin

repeat

While (S1<20)

begin

P(A);

Produce A;

S1=S1+1;

V(A);

End

Forever

end

Product B: 部门B

begin

repeat

While (S2<20)

begin

P(B);

Produce B;

S2=S2+1;

V(B)

End

Forever

end

Procedure:

Assembly: 装配部门

begin

repeat

While (S1>0&&S2>0)

begin

P(A);

P(B);

S1=S1-1

S2=S2-1

Assembly A and B;

V(A);

V(B);

End

forever

end

Cobegin

product A;

product B;

Assembly;

coend



四、银行家算法

在银行家算法中，若出现下述的资源分配情况：

Process	Allocation	Need	Available
p0	1, 0, 2, 3	7, 6, 6, 1	3, 0, 1, 4
p1	2, 1, 0, 1	6, 6, 5, 7	
p2	1, 0, 4, 1	3, 5, 1, 5	
p3	0, 3, 2, 1	3, 1, 2, 8	
p4	3, 2, 1, 0	3, 2, 2, 5	
p5	0, 1, 2, 6	2, 0, 1, 2	

当前状态安全吗？

若进程p3提出请求Request(2, 0, 1, 1)后，系统能否将资源分配给它？为什么？

Available的变化

p5⇒3, 1, 3, 10



p3⇒3, 4, 5, 11



p4⇒6, 6, 6, 11



p1⇒8, 7, 6, 12



p0⇒9, 7, 8, 15



p2⇒10, 7, 12, 16

(1) 安全！

	need	Available
P3⇒	1, 1, 1, 7	1, 0, 0, 3

如果满足 (2, 0, 1, 1) 请求，则所有其它请求再也无法满足，造成死锁。

五、存储分配与地址变换

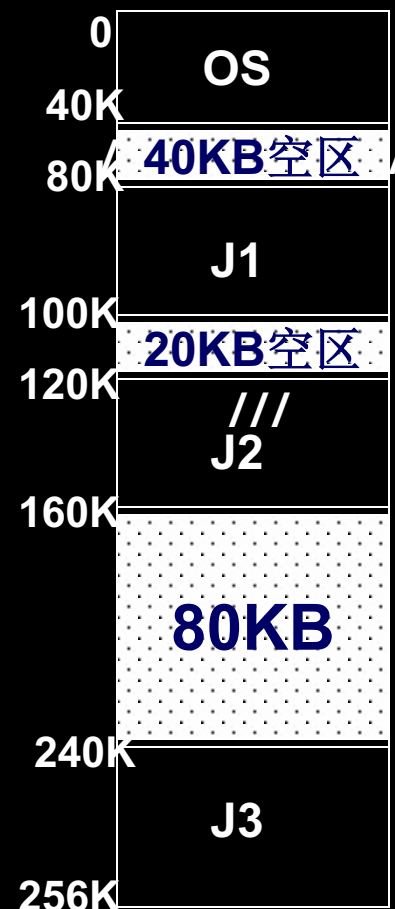
某一采用分段虚拟存储管理的系统, 假定:

(1). 系统提供有序对虚拟字节地址 $v=(s,d)$, 其中 s 是被访问的虚地址所在的段号, d 是它在该段内的偏移量.

(2). 段表格式如下:

段号	段长 (字节)	是否在内存 (y或n)	内存起址 (字节)

(3). 内存物理存储的当前分区状态如图:

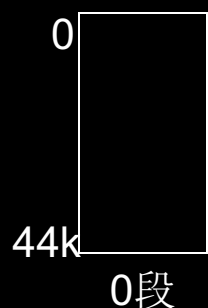


其中带斜线者为空闲区

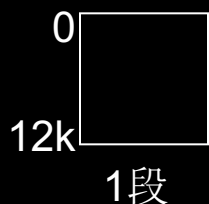
五续一

(4). 系统采用**最佳适应**的空闲区分配算法.

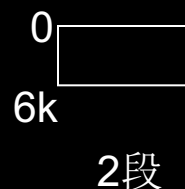
现在调度进程要调度一个有下述逻辑结构的进程到内存



主程序段



子程序段



数据段

调度进程依段号从小到大的顺序为该进程分配内存, 并设法将当前段全部装入内存.

请完成:

1. 填写该进程相应的段表信息.
 2. 图示虚拟地址 v 的再定位过程.
 3. 分别求出主程序段与数据段中字节地址4K所对应的物理地址.
 4. 画出本次调度后的内存分区状态图.
- (注: 本题目不考虑淘汰其它进程的分段)

续二

● 段表地址寄存器

有效地址

0

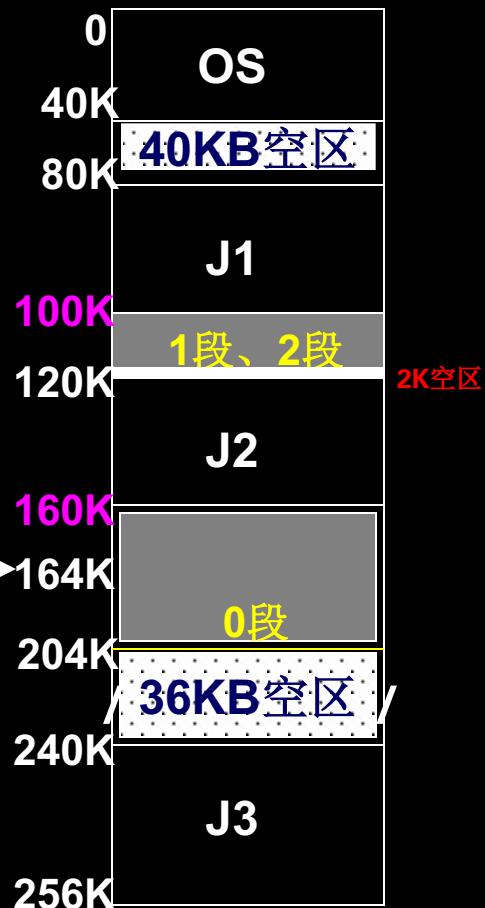
4K

段号	段长 (字节)	是否在内存 (y或n)	内存起址 (字节)
0	44K	y	160
1	12K	y	100
2	6K	y	112

+



其中带斜线者为空闲区



六、磁盘调度

应用SSTF (最短寻道时间shortest-seek-time-first)调度策略，某些进程可能永远不能被调度到。如右表所示的例子中，我们假定每当进程9（要求读出磁道376上的信息）的请求得到服务之前的某段时间，系统又接收一个请求流，而且这些请求所要移动磁头的距离总小于达到磁道376所移动的距离，因而，进程9和3将永远得不到服务。

试设计一种磁盘访问调度算法，以确保不会发生诸如上例的“饥饿”现象。

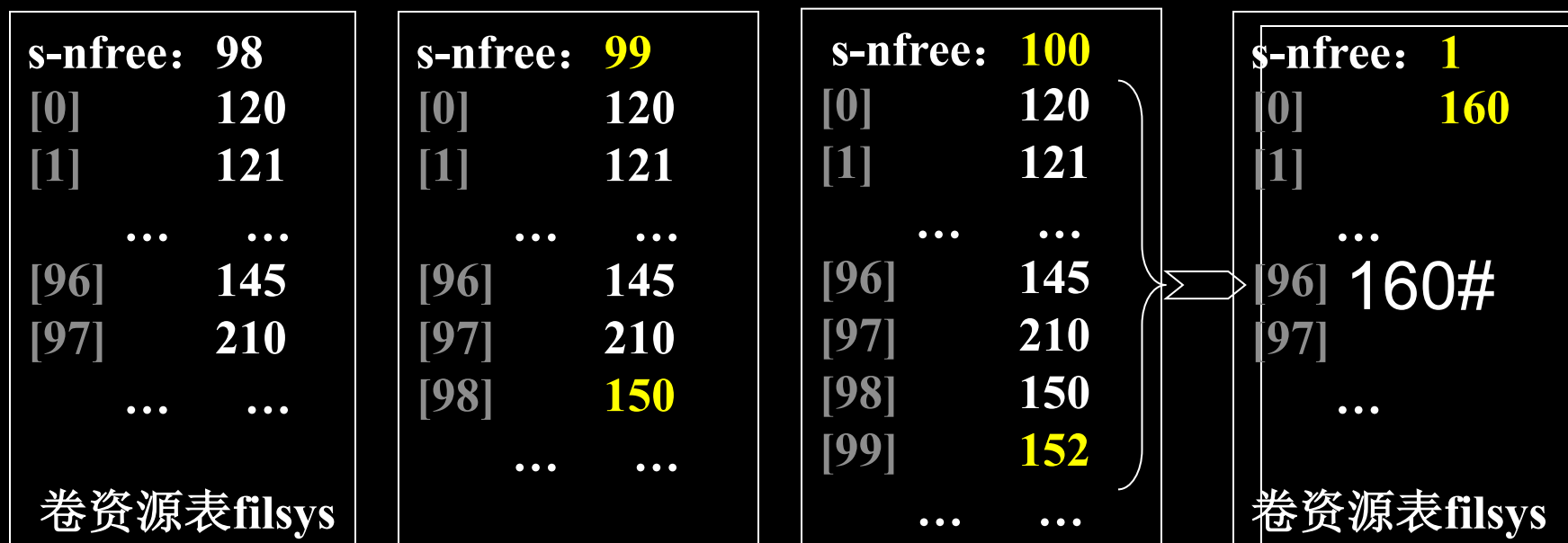
进程号	磁道号	移动磁道数
7	134	0
14	192	58
23	205	13
22	56	149
29	40	16
32	29	11
4	19	10
12	19	0
34	18	1
17	3	15
9	376	373
3	396	20

答：无饥饿现象的磁盘调度算法有FCFS、扫描算法等等。



七、磁盘空闲块的成组分配算法

参看下图，现有某一进程的文件要释放三个物理块，其块号为150#，152#，160#，试给出其释放过程和释放后的卷资源表filsys的状况。其后，又有一个文件要求分配4个空闲块，试给出其分配过程和分配后的filsys状况：



七 (续)

s-nfree: 98

[0] 120

[1] 121

... ..

[96] 145

[97] 210

... ..

卷资源表filsys

210#

s-nfree: 99

[0] 120

[1] 121

... ..

[96] 145

[97] 210

[98] 150

... ..

150#

s-nfree: 100

[0] 120

[1] 121

... ..

[96] 145

[97] 210

[98] 150

[99] 152

... ..

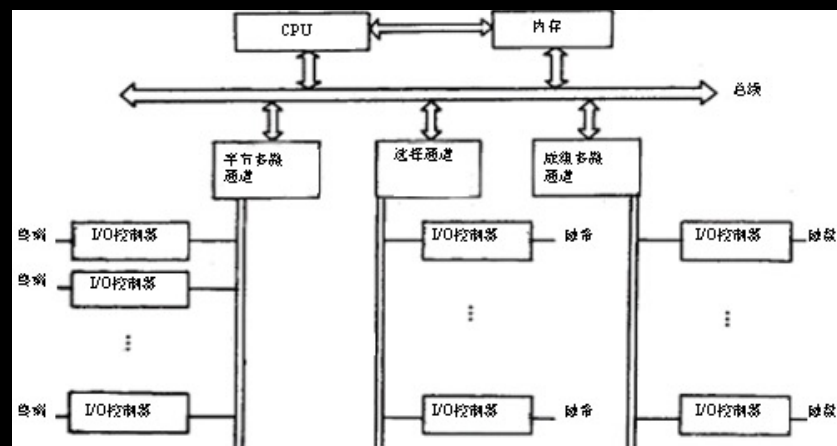
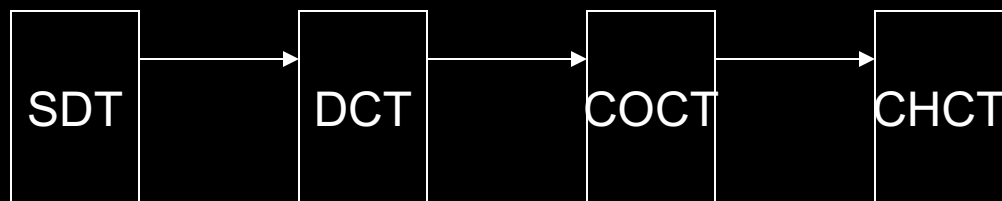
152#

160#

设备管理概念题

1、用于设备分配的数据结构有哪些？它们之间的关系是什么？

答：整个系统有一张SDT，每个设备有一张DCT，每个控制器有一张COCT，每个通道有一张CHCT。



2、什么是I/O控制？它的主要任务是什么？

答：输入输出处理过程分为两个阶段，即：

- + 用户I/O请求→分配设备→启动设备进行I/O操作
- + 响应设备中断→设备中断处理

从处理I/O请求开始到设备中断处理结束，这一总过程称为I/O控制。

设备管理概念题续

3、I/O控制可用哪几种方式实现？各有什么优缺点？

答：I/O控制过程可用三种方式实现：作为请求I/O操作的进程实现；作为当前进程的一部分实现；由专门的系统进程（I/O进程）完成。

第一种方式请求对应I/O操作的进程能很快占据处理机，但要求系统和I/O操作的进程应具有良好的实时性。第二种方式不要求系统具有高的实时性，但I/O控制过程要由当前进程负责。第三种方式增加了一个额外的进程开销，但用户进程不用处理I/O控制过程。

4、设备驱动程序是什么？为什么要设备驱动程序？用户进程怎样使用驱动程序？

答：设备驱动程序是驱动外部物理设备和相应DMA控制器或I/O控制器等器件，使之可以直接和内存进行I/O操作的子程序的集合。它们负责设置相应设备有关寄存器的值，启动设备进行I/O操作，指定操作的类型和数据流向等。

设备驱动程序屏蔽了直接对硬件操作的细节，为编程者提供操纵设备的便利接口。

用户进程通过调用设备驱动程序提供的接口来使用设备驱动程序。

八、问题参考

- 操作系统的类型
 - + 每种操作系统追求的目标是什么？
 - + 分析操作系统的几种观点强调的侧重点各是什么？
 - + 分布式系统与网络系统之间的关系
- 什么是管态和目态？用户程序如何进入管态执行？
- 什么是特权指令？用户程序可以执行特权指令吗？
- 什么是系统调用？
- 什么是进程同步与互斥，什么是进程同步原语？原语的含义？
- 什么是临界区和临界资源？
- 什么是PCB？什么是进程？什么是进程调度？
- 什么是多道程序设计？什么是死锁以及死锁产生的必要条件是什么？
- P、V操作的定义？实际物理意义？
- 管程的特点
- 死锁定义、进程资源图的化简方法
- 预防死锁、避免死锁的方法有哪些？
- 什么是高级调度、中级调度和低级调度？
- 解释DCT、FCB、JCB
- 目录项和FCB的区别是什么？
- 简述页式存储管理的实现原理及其地址变换过程。
- 在分区式存储管理中硬件提供那些支持？



问题参考（续）

- 在页式存储管理中，硬件提供哪些支持？
- 什么是虚拟存储管理？虚拟存储器的大小受什么限制？
- 那些存储管理技术可用于虚存管理？
- 什么是虚拟机？
- 什么是作业？什么是作业控制块
- 进程的基本状态有哪些，状态之间的转换的原因是什么？
- 什么是设备管理中的主设备号和次设备号？
- 中断处理应包括哪些主要步骤？
- 中断与陷阱的区别？
- 文件的定义
- 文件的逻辑结构有哪些？
- 文件的物理组织有哪些？
- 描述实现文件的系统的层次结构。
- 何谓文件的共享、保护和保密？
- 何谓实时信息处理系统和实时控制系统？
- 何谓硬实时？何谓软实时？
- 网络与分布式操作系统有何区别？
- 进程与程序有何区别？
- 进程与线程有何区别？

