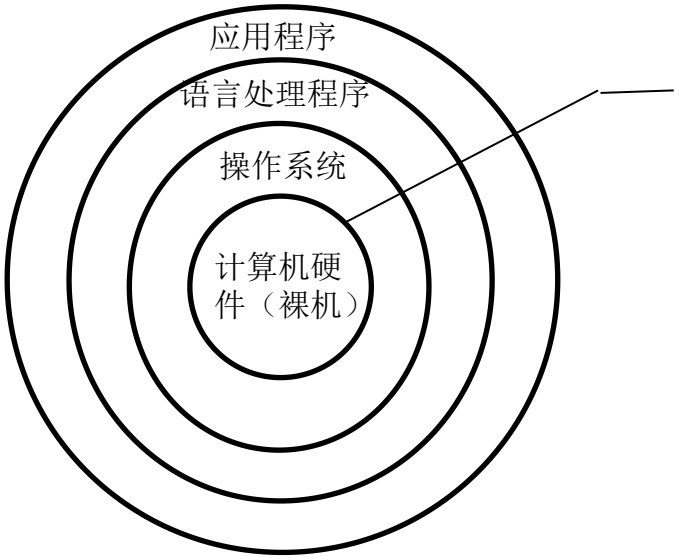
系统架构设计师

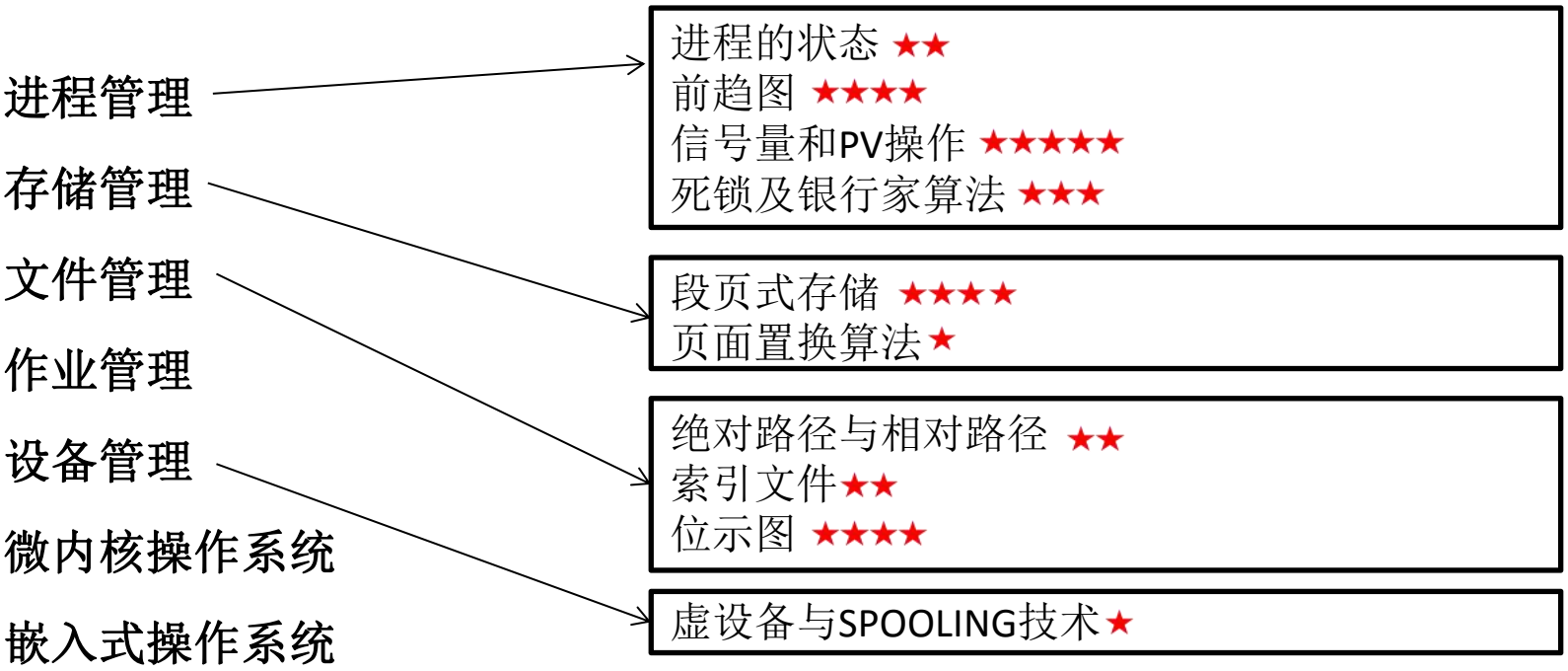
DESIGNER:王川林

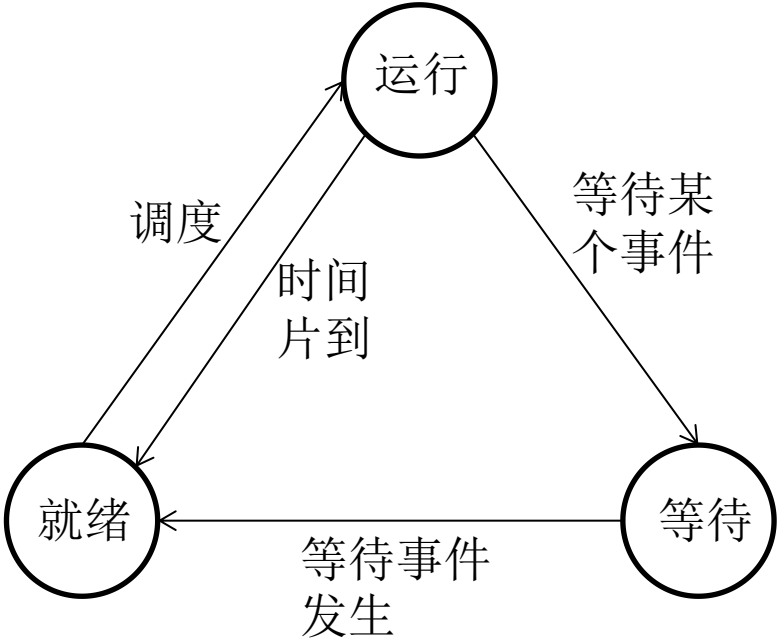
操作系统

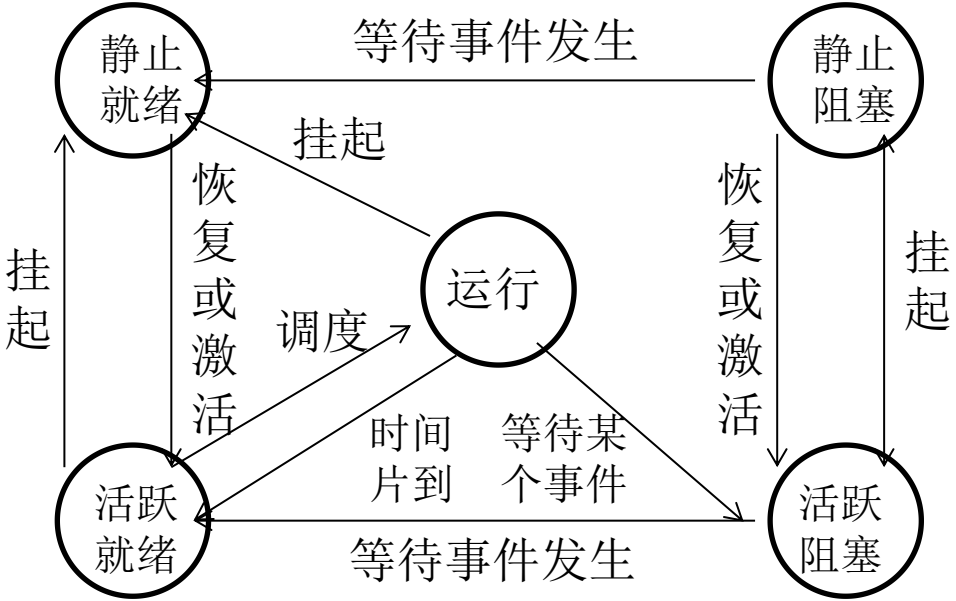


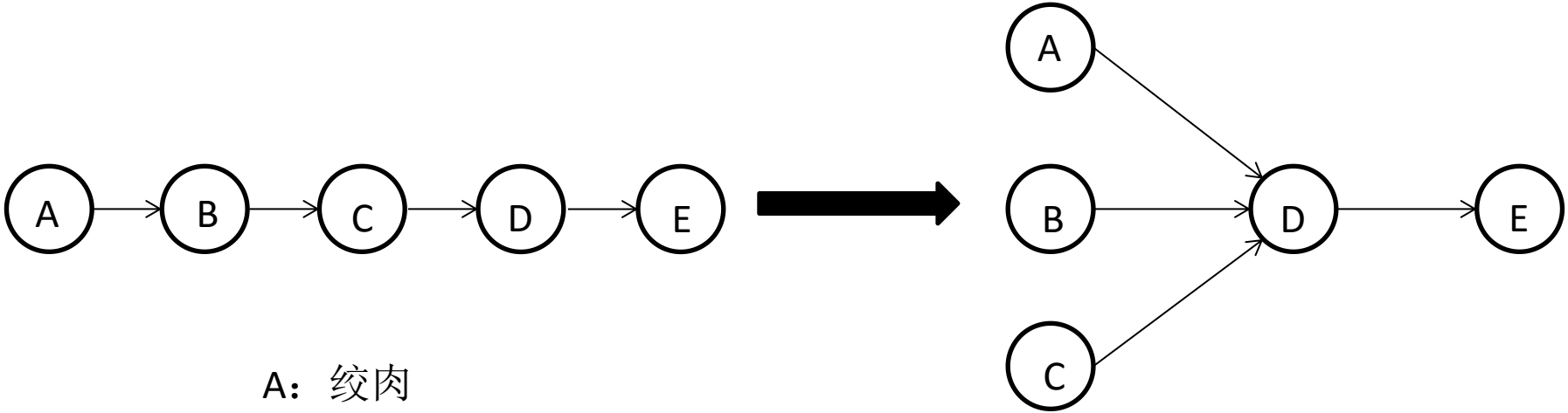
- 管理系统的硬件、软件、数据资源
- 控制程序运行
- 人机之间的接口
- 应用软件与硬件之间的接口

- 进程管理
- 存储管理
- 文件管理
- 作业管理
- 设备管理

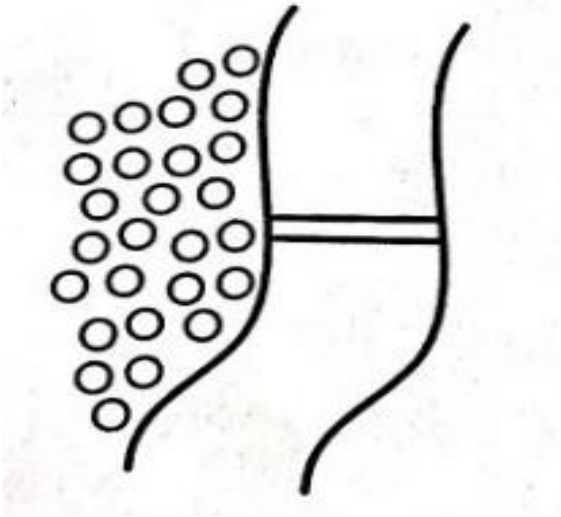

进程管理
存储管理
文件管理
作业管理
设备管理
微内核操作系统
嵌入式操作系统
进程的状态 ★★
前趋图 ★★★★
信号量和PV操作 ★★★★★
死锁及银行家算法 ★★★
段页式存储 ★★★★
页面置换算法 ★
绝对路径与相对路径 ★★
索引文件★★
位示图 ★★★★
虚设备与SPOOLING技术★


运行
调度
时间
片到
等待某
个事件
就绪
等待
等待事件
发生

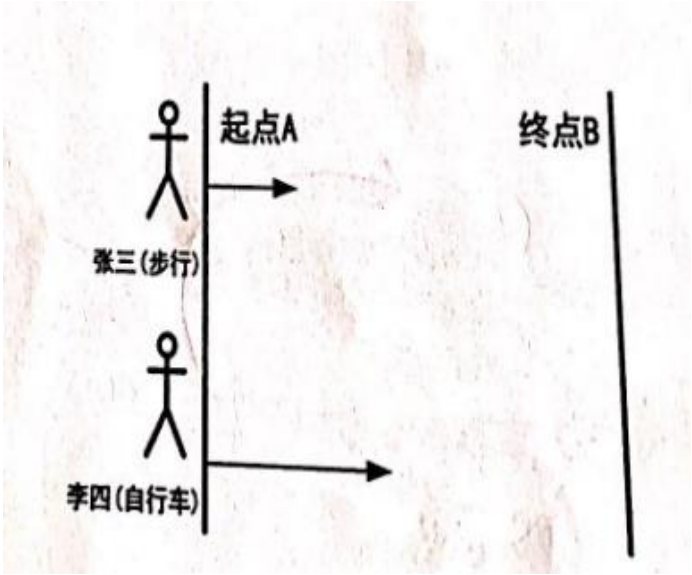

静止
就绪
等待事件发生
静止
阻塞
挂起
恢复
或
激
活
挂
起
运行
调度
恢复
或
激
活
挂
起
时间
片到
等待某
个事件
活跃
就绪
活跃
阻塞
等待事件发生



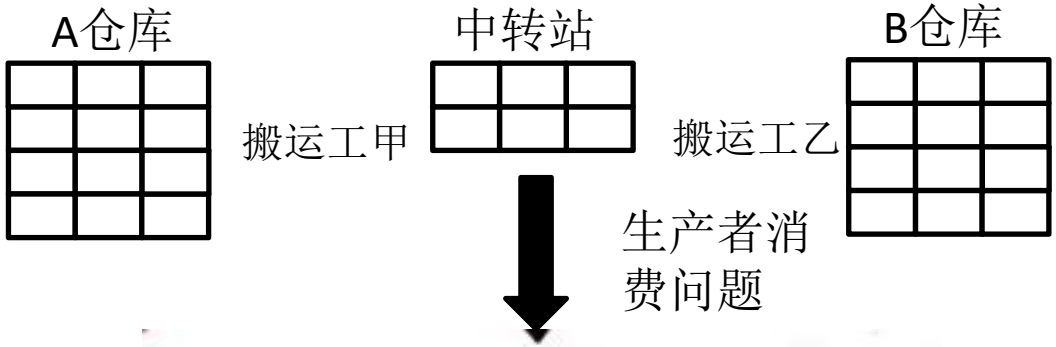
A：绞肉
B：切葱末
C：切姜末
D：搅拌
E：包饺子



互斥：如千军万马过独木桥



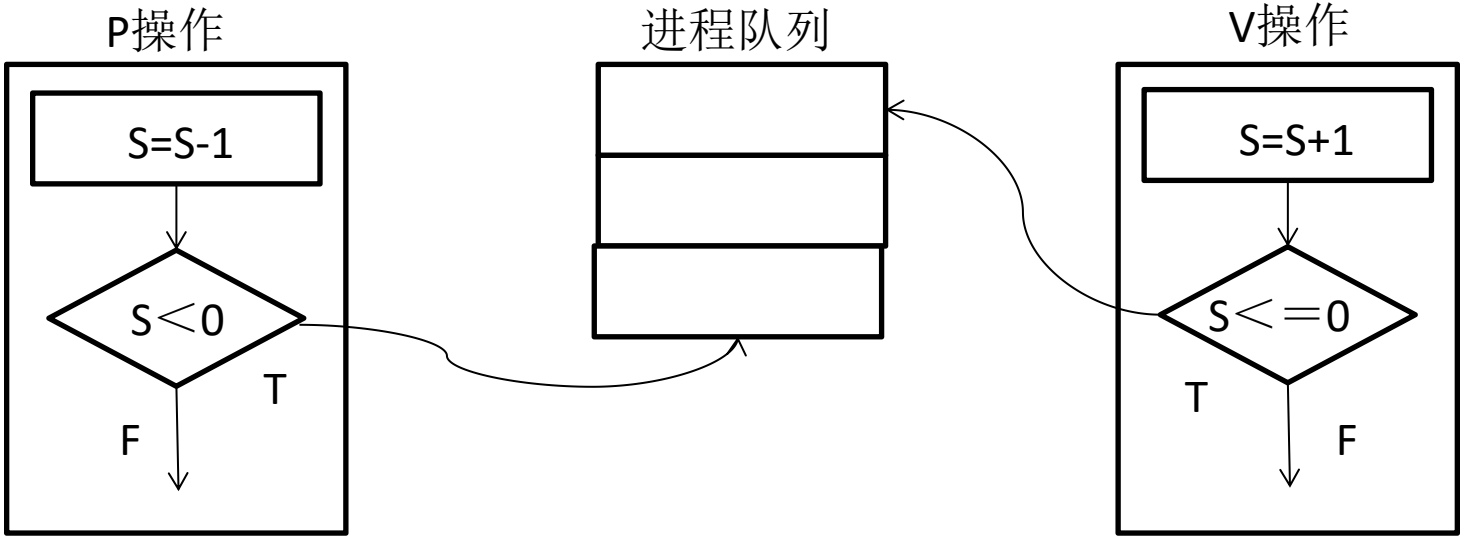
同步：速度有差异，在一定情况停下等待



单缓冲区情况

多缓冲区情况

- 临界资源：诸进程间需要互斥方式对其进行共享的资源，如打印机、磁带机等
- 临界区：每个进程中访问临界资源的那段代码称为临界区
- 信号量：是一种特殊的变量

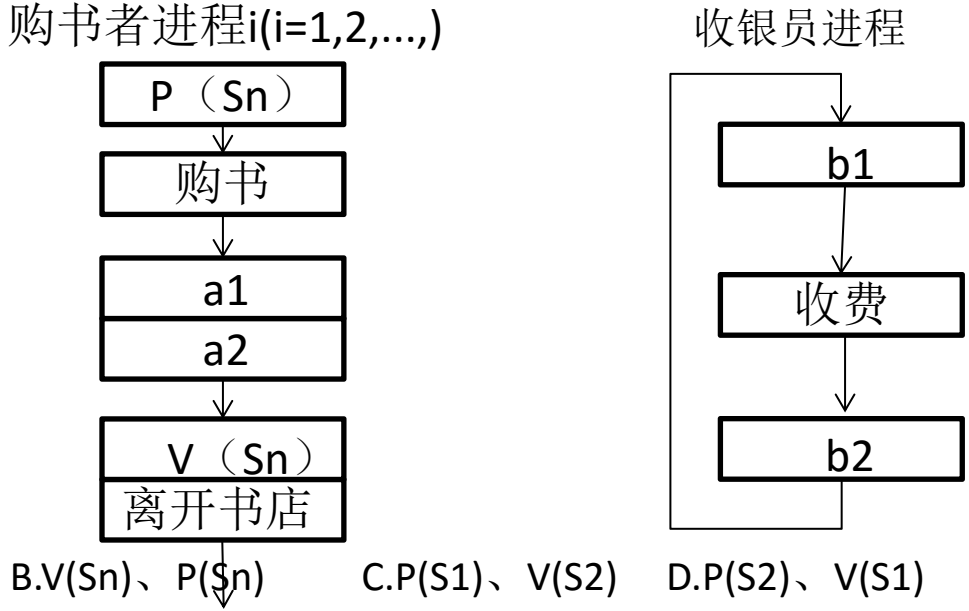


生产者：
生产一个产品；
P（s1）；
送产品到缓冲区；
V（s2）；

消费者：
P（s2）；
从缓冲区取产品；
V（s1）；
消费产品；

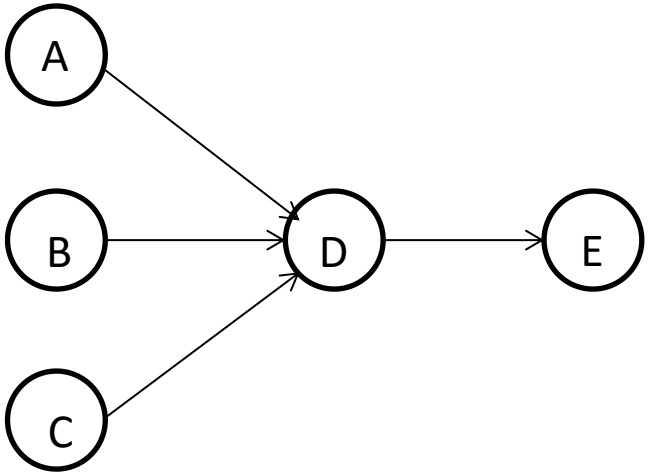
S1初值为1，S2初值为0

某书店有一个收银员，该书店最多允许n个购书者进入。将收银员和购书者看作不同的进程，其工作流程如下图所示。利用PV操作实现该过程，设置信号量S1、S2和Sn，初值分别为0，0，n。则图中a1和a2应填入_(1)_，图中b1和b2应填入_(2)_。



（1）A.V(S1)、P(S2)　B.V(Sn)、P(Sn)　C.P(S1)、V(S2)　D.P(S2)、V(S1)

（2）A.P(Sn)、V(S2)　B.V(Sn)、P(S2)　C.P(S1)、V(S2)　D.P(S2)、V(S1)



A：绞肉
B：切葱末
C：切姜末
D：搅拌
E：包饺子

Sa=0;
Sb=0;
Sc=0;
Sd=0;

进程A：
绞肉；
V(Sa);

进程B：
切葱末；
V(Sb);

进程C：
切姜末；
V(Sc);

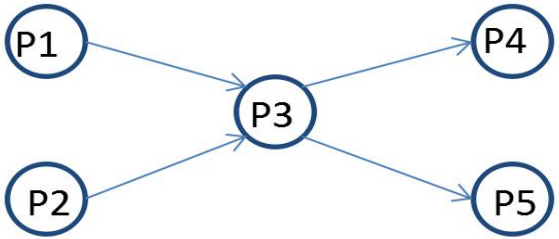
进程D：
P(Sa);
P(Sb);
P(Sc);
搅拌；
V(Sd);

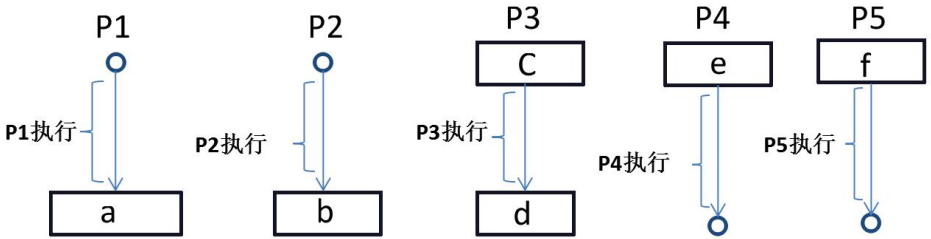
进程E：
P(Sd)；
包饺子;

DESIGNER:

进程管理—PV操作

进程P1、P2、P3、P4、P5的前趋图如下图所示。若用PV操作的控制进程并发执行的过程，则需要设置4个信号量S1、S2、S3和S4，且信号量初值都等于零。图中a和b应分别填写（1）。C和d应分别填写（2），e和f应分别填写（3）。





（1）A. P（S1）和P（S2）　　B. P（S1）和V（S2）

C. V（S1）和V（S2）　　D.V（S1）和P（S2）

（2）A. P（S1）、P（S2）和V（S3）、V（S4）

B. P（S1）、P（S2）和P（S3）、P（S4）

C. V（S1）、V（S2）和P（S3）、P（S4）

D.V（S1）、V（S2）和V（S3）、V（S4）

（3）A.P（S3）和P（S4）　　B.P（S3）和V（S4）

C.V（S3）和V（S4）　　D.V（S3）和P（S4）

DESIGNER:

进程管理—PV操作

假设某系统采用非抢占式优先级调度算法，若改系统有两个优先级相同的进程P1和P2，各进程的程序段如下所示，若信号量S1和S2的初值都为0.进程P1和P2并发执行后a、b和C的结果分别为：a=（4），b=（5），C=（6）。

P1程序段

```
begin(
    a:=1;
     a:=a+1;
     V(S1);
     C:=a+5;
      P(S2);
      a:=a+c;
      }
     end
```

P2程序段

```
begin(
    b:=2
     b:=b+1;
     P(S1);
     b:=a+b
      V(S2);
     C:=b+c;
      }
     end
```

（4）A.9　　B.12　　C.13　　D.14

（5）A.5　　B.6　　C.9　　D.10

（6）A.4　　B.6　　C.12　　D.13

DESIGNER:

进程管理—PV操作

P1程序段

```
begin{
    a:=1 ;
    a:=a+1 ; // a=2
    V ( S1 ) ;
    c:=a+5 ; // c=7
    P ( S2 ) ; //唤醒
    a:=a+c ; // a=14
  }
end
```

P2程序段

```
begin{
    b:=2 ;
    b:=b+1 ; // b=3
    P ( S1 ) ; //唤醒
    b:=a+b ; // b=5
    V ( S2 ) ;
    c:=b+c ; // c=12
  }
end
```

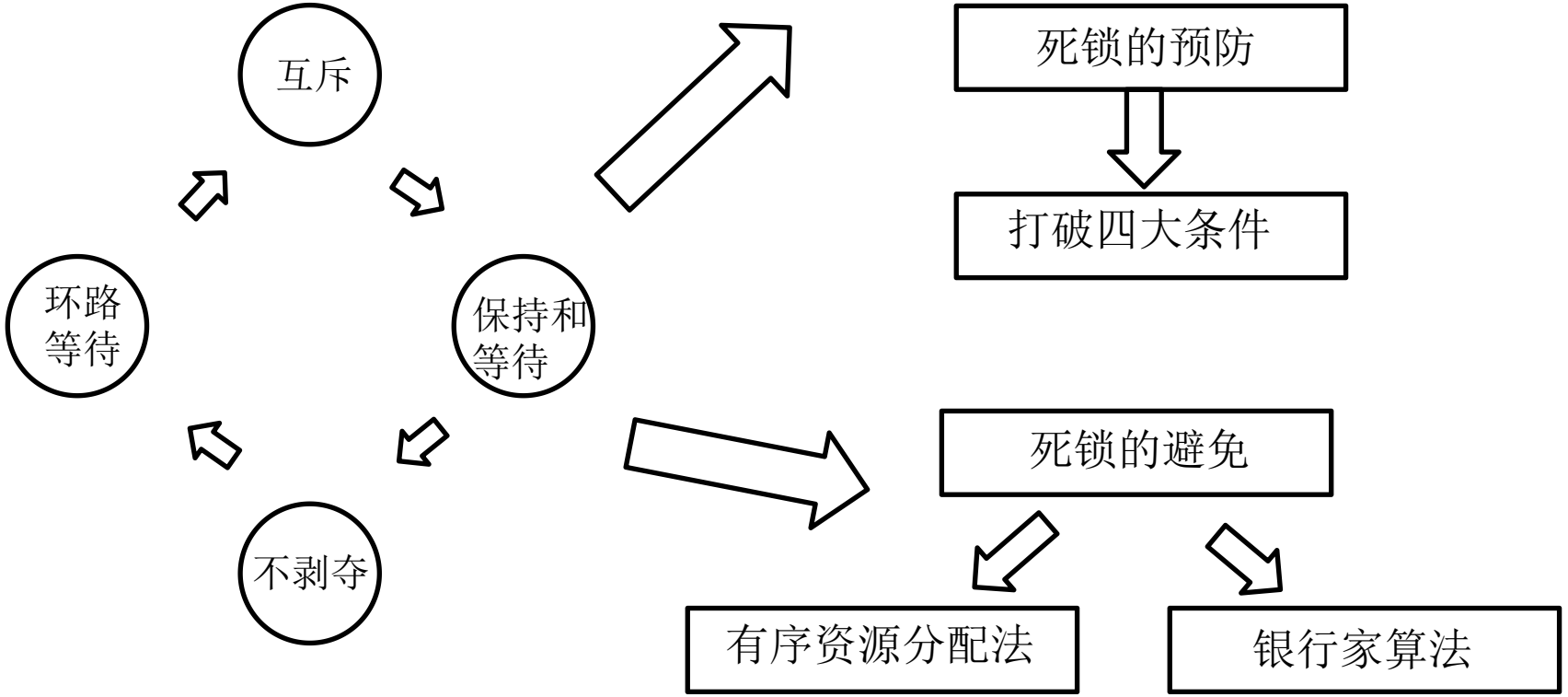
DESIGNER:

进程管理—死锁问题

进程管理师操作系统的核心，但如果设计不当，就会出现死锁的问题。如果一个进程在等待一件不可能发生的事，则进程就死锁了。而如果一个或多个进程产生死锁，就会造成系统死锁。

例：系统有3个进程：A、B、C。这3个进程都需要5个系统资源。如果系统至少有多个资源，则不可能发生死锁。

进程A	进程B	进程C


互斥
保持和等待
不剥夺
环路等待
死锁的预防
打破四大条件
死锁的避免
有序资源分配法
银行家算法

银行家算法：分配资源的原则

- 当一个进程对资源的最大需求量不超过系统中的资源数时可以接纳该进程
- 进程可以分期请求资源，单请求的总数不能超过最大需求量
- 当系统现有的资源不能满足进程尚需资源数时，对进程的请求可以推迟分配，但总能使进程在有限的时间里得到资源

DESIGNER:

进程管理—银行家算法

银行家算法例子：

假设系统中有三类互斥资源R1、R2、R3，可用资源分别是9、8、5。在T0时刻系统中有P1、P2、P3、P4和P5五个进程，这些进程对资源的最大需求量和已分配资源数如下所示，如果进程按____序列执行，那么系统状态是安全的。

进程 \ 资源	最大需求量			已分配资源数		
	R1	R2	R3	R1	R2	R3
P1	6	5	2	1	2	1
P2	2	2	1	2	1	1
P3	8	1	1	2	1	0
P4	1	2	1	1	2	0
P5	3	4	4	1	1	3

供选择的答案：

A.P1→P2→P4→P5→P3

B.P2→P4→P5→P1→P3

C.P2→P1→P4→P5→P3

D.P4→P2→P5→P1→P3

DESIGNER:

进程管理—银行家算法

首先求剩下的资源数：

R1=9（1+2+2+1+1）=2

R2=8（2+1+1+2+1）=1

R3=5（1+1+3）=0

还需资源数表

进程 \ 资源	最大需求量 R1	R2	R3	已分配资源数 R1	R2	R3	还需资源数 R1	R2	R3
P1	6	5	2	1	2	1	5	3	1
P2	2	2	1	2	1	1	0	1	0
P3	8	1	1	2	1	0	6	0	1
P4	1	2	1	1	2	0	0	0	1
P5	3	4	4	1	1	3	2	3	1

求序列P 2→P 4→P 1→P 3是否安全：

进程序列P 2→P 4→P 5→P 1→P 3运行分析表

进程 \ 资源	现有资源 R1	R2	R3	需要资源 R1	R2	R3	已经分配 R1	R2	R3	现有+已经分配 R1	R2	R3	完成
P2	2	1	0	0	1	0	2	1	1	4	2	1	True
P4	4	2	1	0	0	1	1	2	0	5	4	1	True
P5	5	4	1	2	3	1	1	1	3	6	5	4	True
p1	6	5	4	5	3	1	1	2	1	7	7	5	True
P3	7	7	5	6	0	1	2	1	0	9	8	5	True

DESIGNER:

进程管理—银行家算法

求序列P 2→P 1→P 4→P 5→P 3是否安全：

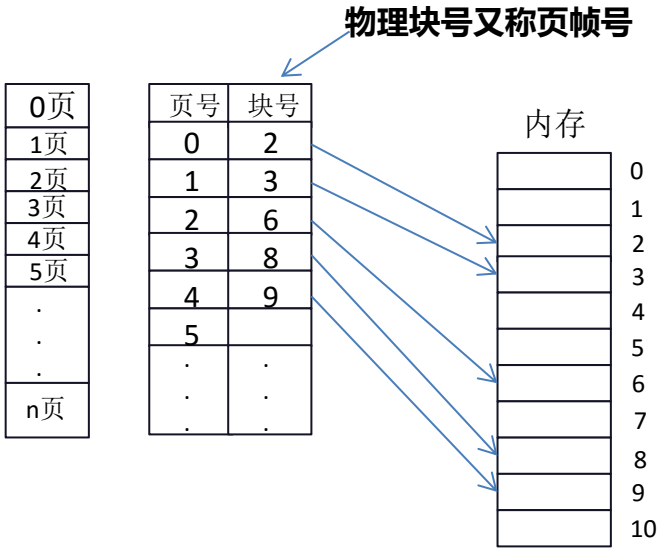
进程序列P 2→P 1→P 4→P 5→P 3运行分析表

资源 / 进程	现有资源 R1 R2 R3	需要资源 R1 R2 R3	已经分配 R1 R2 R3	现有+已经分配 R1 R2 R3	完成
P2	2 1 0	0 1 0	2 1 1	4 2 1	True
P1	4 2 1	5 3 1	1 2 1		

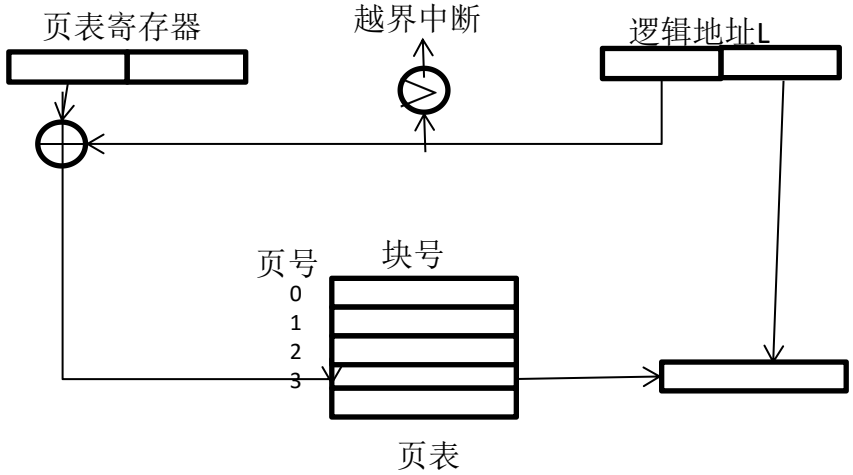
这时候，我们发现进程P1需要R1资源为5，我们能提供的R1资源为4，所以序列无法进行下去，为不安全序列。

DESIGNER:

操作系统—页式存储组织



高级程序语言使用逻辑地址；运行状态，内存中使用物理地址



★优点：利用率高，碎片小，分配及管理简单

★缺点：增加了系统开销；可能产生抖动现象

DESIGNER:
练习题

进程P有6个页面，页号分别为0~5，页面大小为4K，页面变换表如下所示。表中状态位等于1和0分别表示页面在内存和不在内存。假设系统给进程P分配了4个存储块，进程P要访问的逻辑地址为十六进制5A29H，那么该地址经过变换后，其物理地址应为十六进制（1）；如果进程P要访问的页面4不在内存，那么应该淘汰页号为（2）的页面。

页号	页帧号	状态位	访问位	修改位
0	2	1	1	0
1	3	1	0	1
2	5	1	1	0
3	——	0	0	0
4	——	0	0	0
5	6	1	1	1

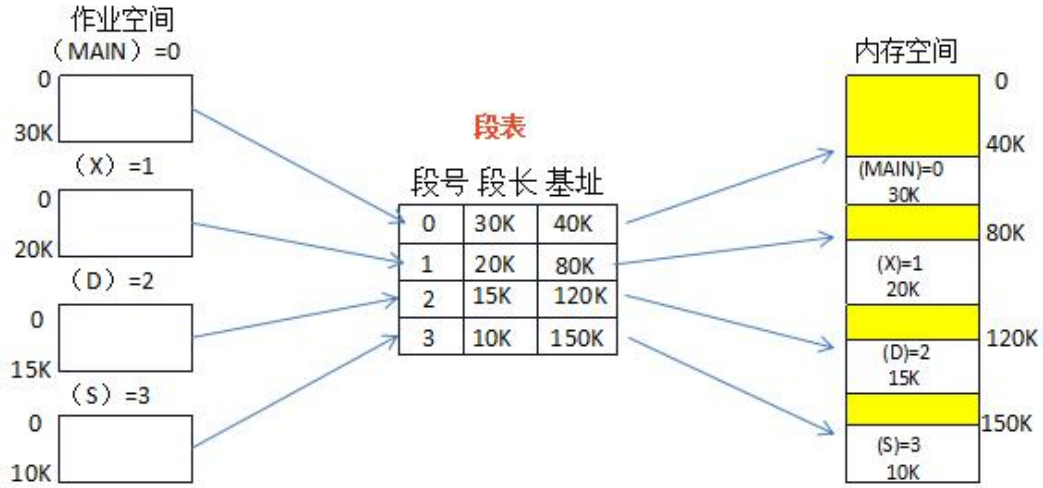
A.1A29H　　B.3A29H　　C.5A29H　　D.6A29H

A.0　　B.1　　C.2　　D.5

DESIGNER:

操作系统—段式存储组织

段式存储：按用户作业中的自然段来划分逻辑空间，然后调入内存，段的长度可以不一样。



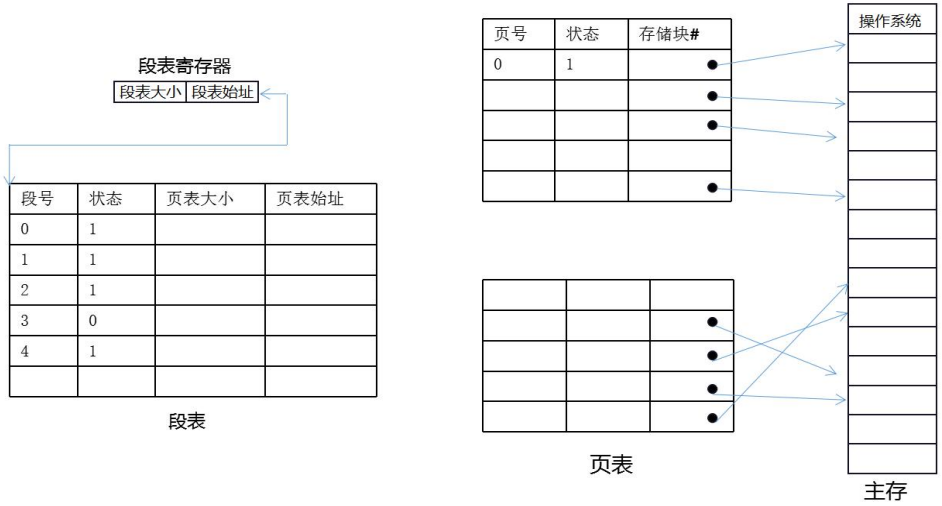
合法段地址：（0, 25K）
非法段地址：（0, 35K）

★ 优点：多道程序共享内存，各段程序修改互不影响
★ 缺点：内存利用率低，内存碎片浪费大

DESIGNER:

操作系统—段页式存储组织

段页式存储：段式与页式的综合体，先分段，再分页。1个程序有若干个段，每个段中可以有若干个页，每个页的大小相同，但每个段的大小不同。



- 优点：空间浪费小、存储共享容易、存储保护容易、能动态连接
- 缺点：由于管理软件的增加，复杂性和开销也随之增加，需要的硬件以及占用的内容也有所增加，使得执行速度大大下降

DESIGNER:

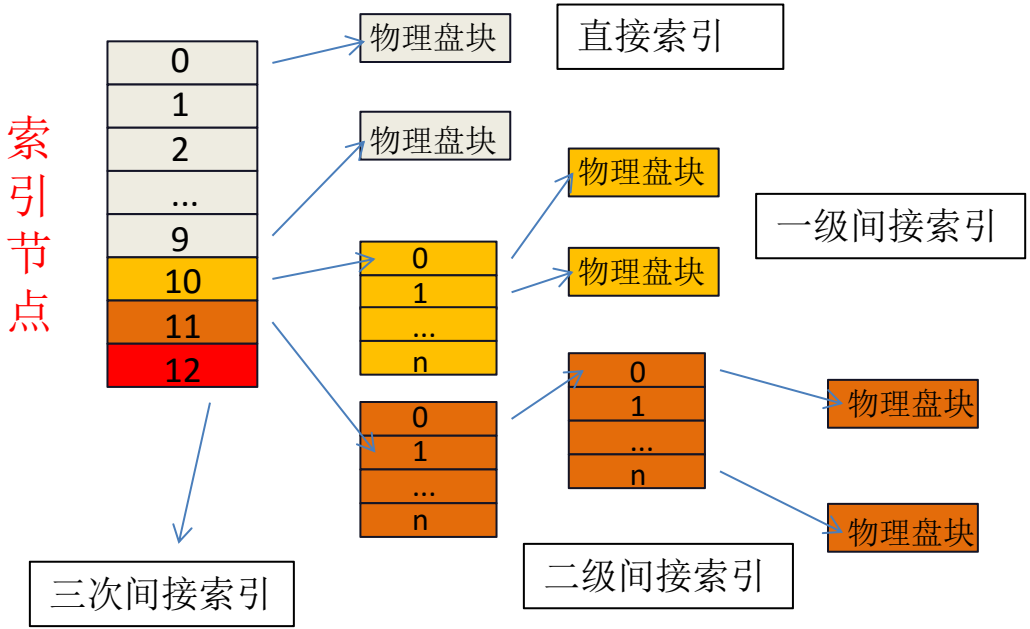
存储管理—快表

- 快表示一块小容量的相联存储器（Associative Memory）,由高速缓存器组成，速度快，并且可以从硬件上保证按内容并行查找，一般用来存放当前访问最频繁的少数活动页面的页号。

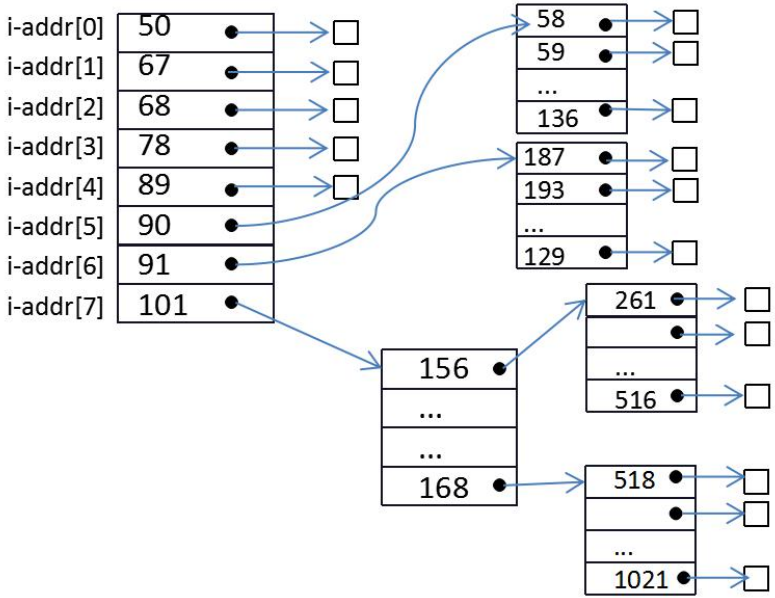
  快表：将页表存于Cache上；慢表：将页表存于内存上。

DESIGNER:

文件管理—索引文件结构



假设文件系统采用索引节点管理，且索引节点有8个地址iaddr[0]~iaddr[7],每个地址项大小为4字节，iaddr[5]和iaddr[6]采用一级间接地址索引，iaddr[7]采用二级间接地址索引。假设磁盘索引块和磁盘数据块大小均为1KB字节，文件File1的索引节点如下图所示。若用户访问文件File1中逻辑块号为5和261的信息，则对应的物理块号分别为_(1)__;101号物理块存放的是_(2)__.



（1）A. 89和90　　B. 89和136

C. 58和187　　D. 90和136

（2）A. File1的信息

B. 直接地址索引表

C. 一级地址索引表

D. 二级地址索引表

DESIGNER:

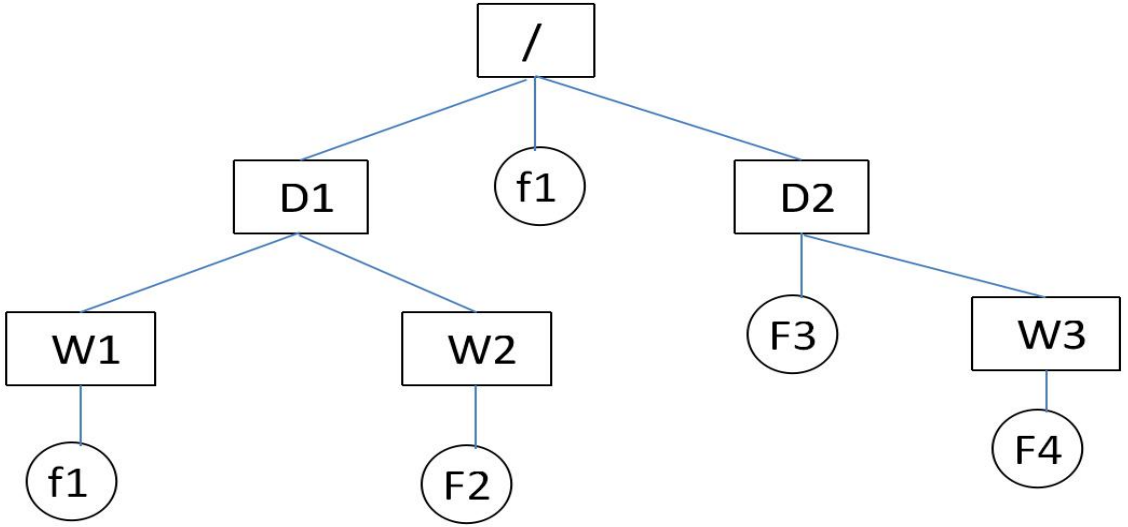
操作系统—文件和树型目录结构

文件属性

- R只读文件属性
- A存档属性
- S系统文件
- H隐藏文件

文件名的组成

- 驱动器号
- 路径
- 主文件名
- 扩展名



- **绝对路径：是从盘符开始的路径。**
- **相对路径：是从当前路径开始的路径。**
- **若当前目前为：D1，要求F2路径，则：绝对路径：/D1/W2/F2，相对路径：W2/F2**

DESIGNER:

文件管理—空闲存储空间的管理

- 空闲区表法（空闲文件目录）
- 空闲链表法
- 位示图法
- 组成链接法

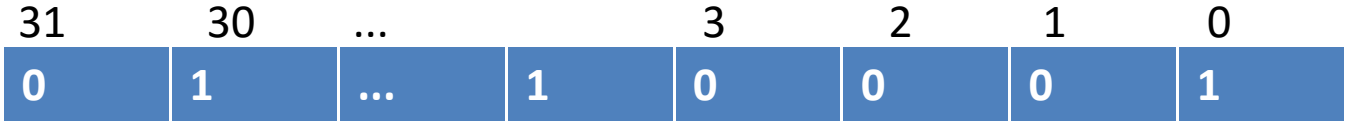
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	0	1	1	1	0	0	1	0	1	1	1	0
1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
2	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
3																
…																
15																

位示图

DESIGNER:

文件管理—试题讲解

某文件管理系统在磁盘上建立了位示图（bitmap）,记录磁盘的使用情况。若磁盘上的物理块依次编号为：0、1、2...，系统中字长为32位，每一位对应文件存储器上的一个物理块，取值0和1分别表示空闲和占用，如下图所示。



假设将4195号物理块分配给某文件，那么该物理块的使用情况在位示图中的第（1）个字中描述；系统应该将（2）。

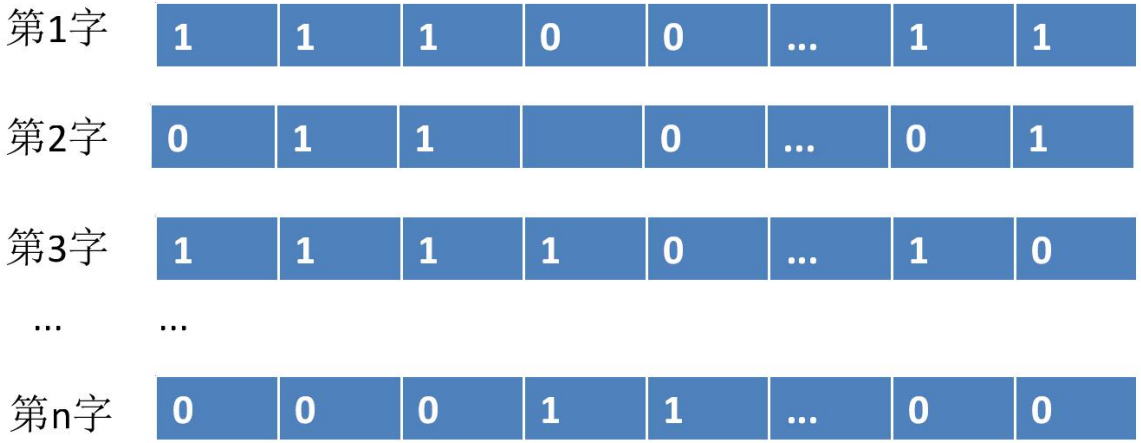
（1）A. 129　　B. 130　　C. 131　　D. 132

（2）A. 该字的第3位置“0”　　B. 该字的第3位置“1”

C. 该字的第4位置“0”　　D. 该字的第4位置“1”

DESIGNER:

文件管理—试题讲解



位示图例

（4195+1）/32=131.125→第132字。

131*32=4192 → 0 - 4191.

第132字中：

第0位置→ 4192　第1位置 → 4193　第2位置 → 4194　第3位置 → 4195

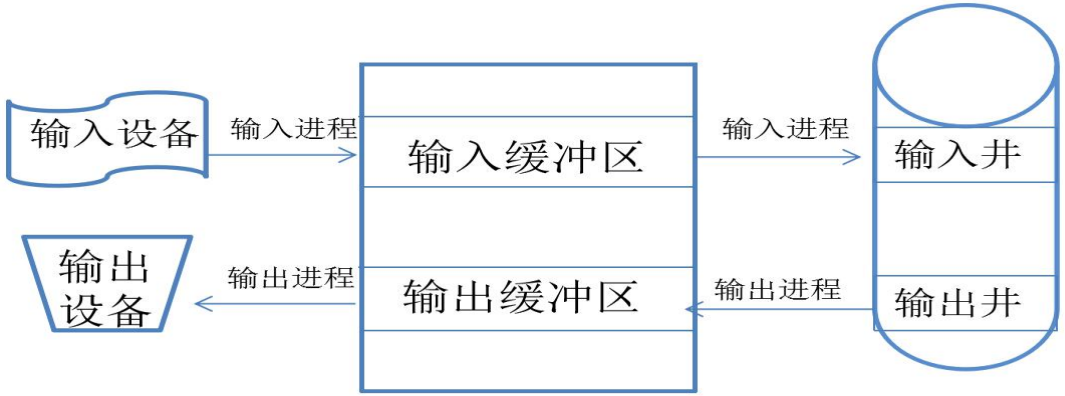
DESIGNER:

设备管理—数据传输控制方式

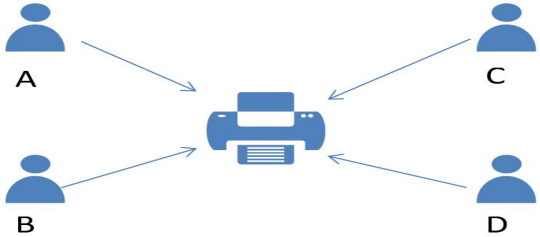
效率越来越高

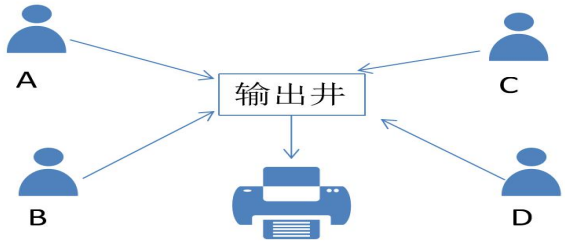
- **程序控制（查询）方式：**分为无条件传送和程序查询方式两种。方法简单，硬件开销小，但 I/O能力不高，严重影响CPU的利用率。
- **程序中断方式：**与程序控制方式相比，中断方式因为CPU无需等待而提高了传输请求的响应速度。
- **DMA方式：**DMA方式是为了在主存与外设之间实现高速、批量数据交换而设置的。DMA方式比程序控制方式与中断方式都高效。
- **通道方式**
- **I/O处理机**

设备管理—虚设备与SPOOLING技术



思考题：A，B，C，D共用一台打印机X，要进行资料打印时，很容易出现“打印机正在使用！”如何处理该问题？





DESIGNER:

微内核操作系统

	实质	优点	缺点
单体内核	将图形、设备驱动及文件系统等功能全部在内核中实现，运行在内核状态和同一地址空间。	减少进程间通信和状态切换的系统开销，获得较高的运行效率	内核庞大，占用资源较多且不易剪裁。系统的稳定性和安全性不好。
微内核	只实现基本功能，将图形系统、文件系统、设备驱动及通信功能放在内核之外	内核精炼，便于剪裁和移植。系统服务程序运行在用户地址空间，系统的可靠性、稳定性和安全性较高。 可用于分布式系统	用户状态和内核状态需要频繁切换，从而导致系统效率不如单体内核。

DESIGNER:
嵌入式操作系统

1.嵌入式操作系统特点：
（1）微型化、（2）代码质量高、（3）专业化、（4）实时性强、（5）可裁减、可配置。
2.实时嵌入式操作系统的内核服务有：异常和中断、计时器、I/O管理。
3.常见的嵌入式RTOS（实时操作系统）VxWorks、RT—Linux、QNX、pSOS.

比较类型	VxWorks	TR-Linux
工作方式	操作系统与应用程序处于同一存储空间	操作系统与应用程序处于不同存储空间
多任务支持	支持多任务（线程））操作	支持多进程、多线程操作
实时性	实时系统	实时系统
安全性	任务间无隔离保护	支持进程间隔离保护
标准API	支持	支持

DESIGNER:王川林

操作系统

THANK YOU