



系统架构设计师

DESIGNER: 王川林
直播课7



第七章 系统配置与性能



- 性能指标 ★
- 阿姆达尔解决方案 ★★
- 性能评价方法 ★★★

主频和CPU时钟周期 (Clock Cycle): 主频又称为时钟频率，时钟周期是时钟频率的倒数。如主频为1GHz，则说明1秒钟有1G个时钟周期，每个时钟周期为 $1000*1000*1000/1G=1ns$ 。

指令周期 (Instruction Cycle): 取出并执行一条指令的时间。

总线周期 (BUS Cycle): 也就是一个访存储器或I/O端口操作所有的时间。

指令周期、总线周期和时钟周期之间的关系: 一个指令周期由若干个总线周期组成，而一个总线周期时间又包含有若干个时钟周期（也可说一个指令周期包含若干个时钟周期）。

MIPS: 每秒处理的百万级的机器语言指令数，主要用于衡量标量机性能。

MFLOPS: 每秒百万个浮点操作，不能反映整体情况，只能反映浮点运算情况，主要用于衡量向量机性能

对时钟频率为**400MHz**的某计算机进行测试，测试程序使用**4**种类型的指令。每种指令的数量及每种指令的指令时钟数（**CPI**）如表所示。该计算机的指令平均时钟数为（ ）；该计算机的运算速度约为（ ）**MIPS**。

指令类型	指令条数	指令时钟数
1	120000	1
2	36000	2
3	24000	4
4	20000	8

A.1.82

B.2.24

C.2.56

D.3.20

A.153.6

B.162.4

C.178.6

D.184.2



对系统中某组件采用某种更快的执行方式，所获得的系统性能的改变程度，取决于该组件被使用的频率，或所占总执行时间的比例。加速比计算公式如下：

$$R = \frac{T_p}{T_i} = \frac{1}{(1 - F_e) + F_e / S_e}$$

其中， T_p 表示不使用改进组件时完成整个任务的时间， T_i 表示使用改进组件时完成整个任务时间。加速比主要取决于两个因素：

(1) 在原有的系统上，能被改进的部分在总执行时间中所占的比例。这个值称为改进比例，记为 F_e ，它总是小于1。

(2) 通过改进的执行方式所取得的性能提高，即如果整个系统使用了改进的执行方式，那么，系统的执行速度会有多少提高，这个值等于在原来的条件下系统的执行时间与使用改进组件后系统的执行时间之比，记为 S_e ，它总大于1。

阿姆达尔 (Amdahl) 定律量化定义了通过改进系统中某个组件的性能，使系统整体性能提高的程度。假设某一功能的处理时间为整个系统运行时间的60%，若使该功能的处理速度提高至原来的5倍，则根据阿姆达尔定律，整个系统的处理速度可提高至原来的（ ）倍。

A. 1.333

B. 1.923

C. 1.5

D. 1.829



- ✓ 时钟频率法：以时钟频率高低衡量速度。
- ✓ 指令执行速度法：表示机器运算速度的单位是MIPS
- ✓ 等效指令速度法：（ Gibson mix, 吉普森混合法 ）：通过各类指令在程序中所占的比例（ W_i ）进行计算得到的。特点：考虑指令比例不同的问题。
- ✓ 数据处理速率法（ PDR ）：PDR值的方法来衡量机器性能，PDR值越大，机器性能越好。 $PDR=L/R$ 特点：考虑CPU+存储
- ✓ 综合理论性能法（ CTP ）：CTP采用MTOPS（ Million Theoretical Operations Per Second，每秒百万次理论运算）表示。CTP的估算方法是，首先算出处理部件每个计算单元的有效计算率，再按不同字长加以调整。得出该计算单元的理论性能，所有组成该处理部件的计算单元的理论性能之和即为CTP。
- ✓ 基准程序法：把应用程序中用得最多、最频繁的那部分核心程序作为评估计算机系统性能的标准程序，称为基准测试程序（ benchmark）。基准程序法是目前一致承认的测试系统性能的较好方法。

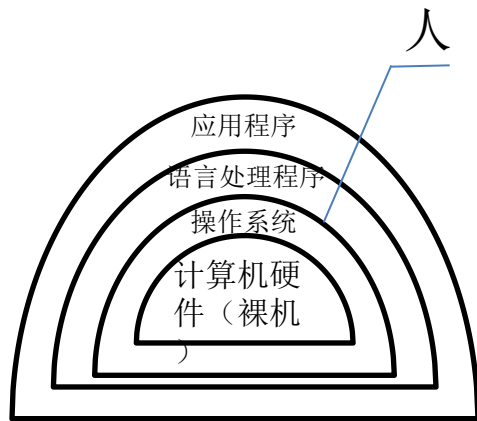
真实的程序->核心程序->小型基准程序->合成基准程序

- ✓ Dhrystone基准程序：它是一个综合性的证书基准测试程序，是为了测试编译器和CPU处理整数指令和控制功能的有效性，人为地选择一些典型指令综合起来形成的测试程序。
- ✓ Linpack基准程序：它是国际上最流行的用于测试高性能计算系统浮点性能的测试。
- ✓ Whetstone基准程序：它是用Fortran语言编写的综合性测试程序，主要由执行浮点运算、功能调用、数组变址、条件转移和超越函数的程序组成。
- ✓ SPEC基准程序一种是测试计算机完成单项任务有多快，称为速度测试；另一种是测试计算机在一定时间内完成多少项任务，称为吞吐率测试。
- ✓ TPC基准程序：TPC (Transaction Processing Council, 事务处理委员会) 基准程序用以评测计算机在事务处理、数据库处理、企业管理与决策支持系统等方面的性能。该基准程序的评测结果用每秒完成的事务处理数TPC来表示。
 - TPC—A基准程序规范用于评价在OLTP环境下的数据库和硬件的性能；
 - TPC—B测试的是不包括网络的纯事务处理量，用于模拟企业计算环境；
 - TPC—C测试的是联机订货系统
 - TPC—D、TPC—H和TPC—R测试的都是决策支持系统，其中TPC—R允许有附加的优化选项；
 - TPC—E测试的是大型企业信息服务系统。
 - TPC—W是基于Web应用的基准程序，用来测试一些通过Internet进行市场服务和销售的商业行为，所以TPC—W可以看作是一个服务的测试标准

以下关于基准测试的叙述中，正确的是（ ）。

- A.运行某些诊断程序，加大负载，检查哪个设备会发生故障
- B.验证程序模块之间的接口是否正常起作用
- C.运行一个标准程序对多种计算机系统进行检查，以比较和评价它们的性能
- D.根据程序的内部结构和内部逻辑，评价程序是否正确

第八章 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

进程的状态 (☆☆)
前趋图 (☆☆☆☆)
信号量与PV操作 (☆☆☆☆☆)
死锁及银行家算法 (☆☆☆)



存储管理

段页式存储 (☆☆☆☆)
页面置换算法 (☆)



文件管理

绝对路径与相对路径 (☆☆)
索引文件 (☆☆)
位示图 (☆☆☆)



作业管理



设备管理

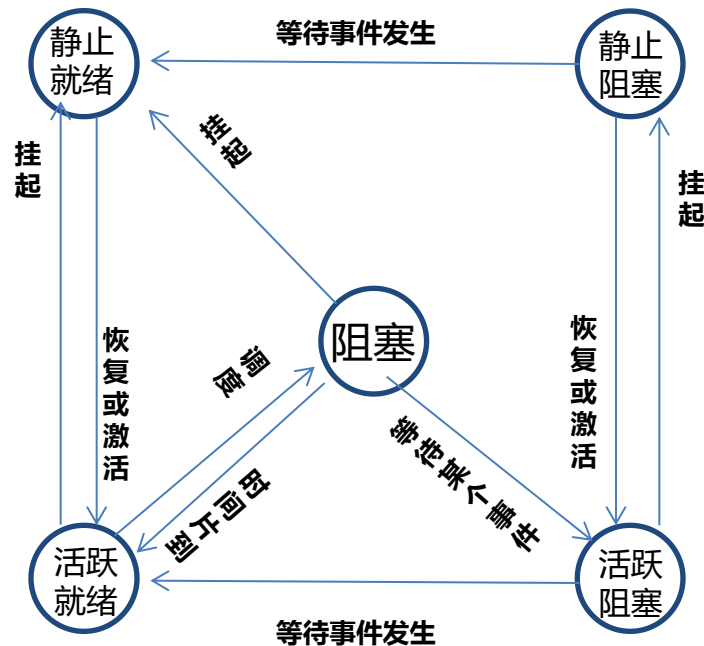
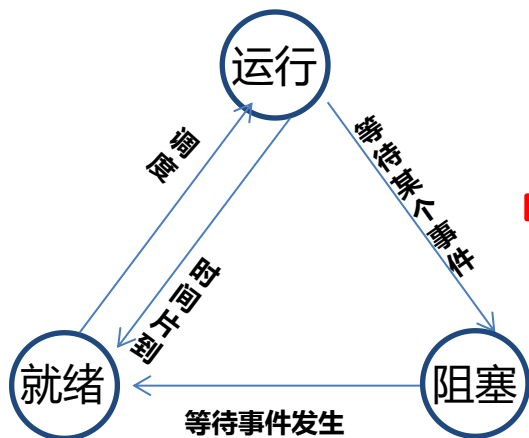
数据传输控制方式 (☆☆)
虚设备与SPOOLING技术 (☆)



微内核操作系统 ★★



嵌入式操作系统



- 进程的五个状态模型包括运行状态、活跃就绪状态、静止就绪状态、活跃阻塞状态和静止阻塞状态。针对下图的进程五态模型，为了确保进程调度的正常工作，(a)、(b)和(c)的状态分别为(1)，并增加一条(2)。

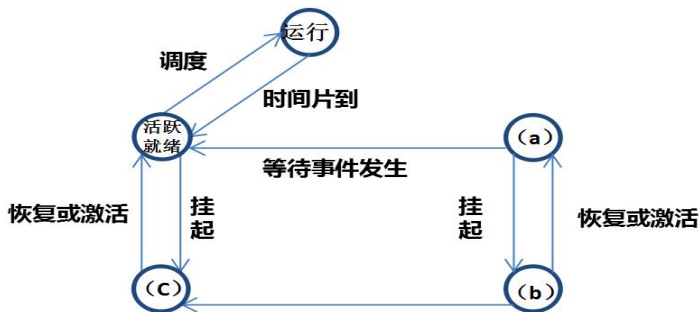


图 进程五态模型

A. 静止就绪、静止阻塞和活跃阻塞

B. 静止就绪、活跃阻塞和静止阻塞

C. 活跃阻塞、静止就绪和静止阻塞

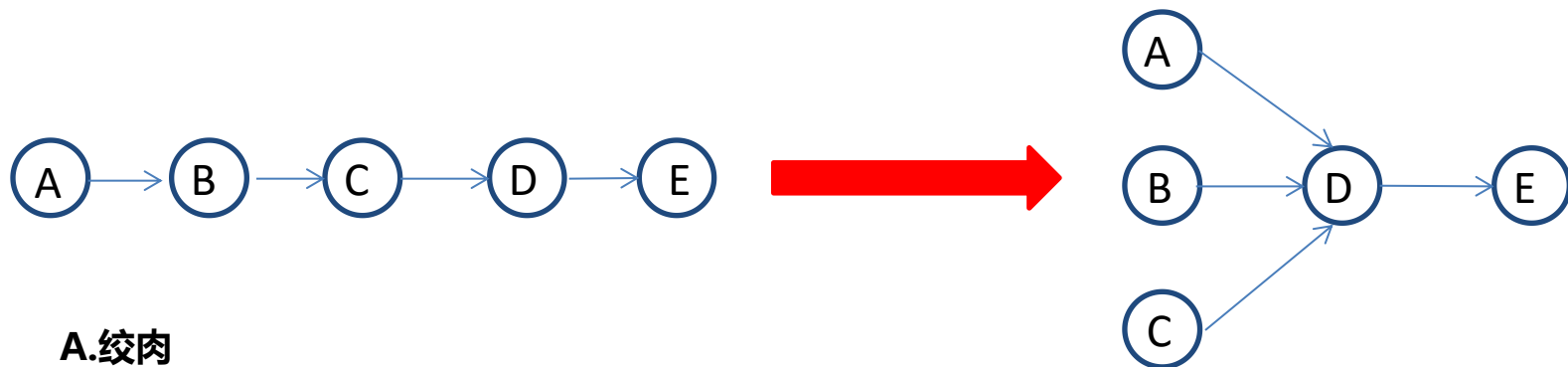
D. 活跃阻塞、静止阻塞和静止就绪

A. “运行” → (a) 的“等待”边

B. “运行” → (b) 的“等待”边

C. (a) → ‘运行’ 的“恢复或激活”边

D. “活动就绪” → (b) 的“等待”边



A.绞肉

B.切葱末

C.切姜末

D.搅拌

E.包饺子

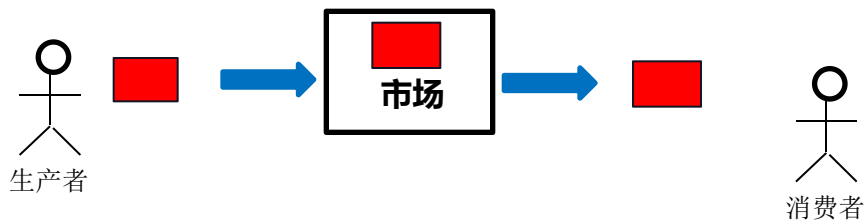
► 如何区分同步与互斥？

互斥：如千军万马过独木桥，同类资源的竞争关系。

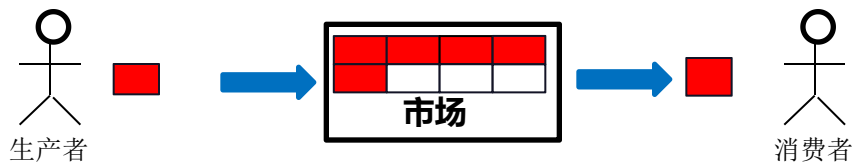
同步：速度有差异，在一定情况停下等待，进程间的协作关系



单缓冲区情况



多缓冲区情况

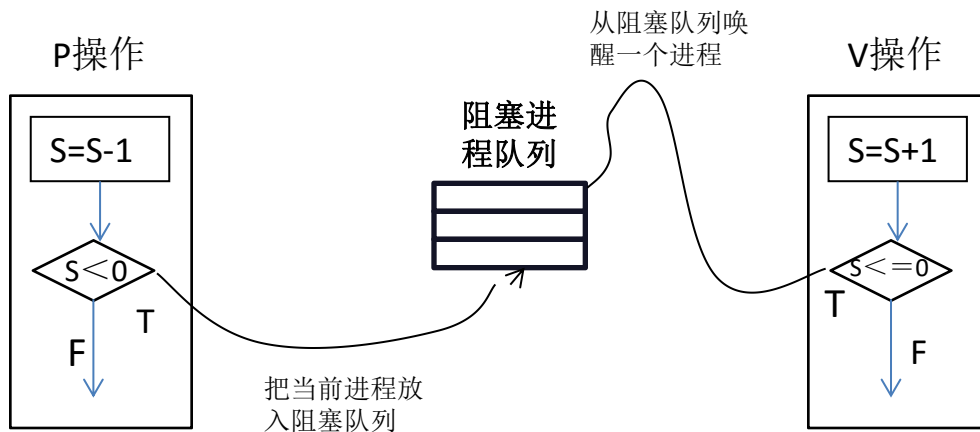


临界资源：诸进程间需要互斥方式对其进行共享的资源，如打印机、磁带机等

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

注：P是荷兰语的Passeren,V是荷兰语的Verhoog.



未使用PV操作：

生产者：

生产一个产品；
送产品到缓冲区；

消费者：

从缓冲区取出产品；
消费产品；



使用PV操作：

生产者：

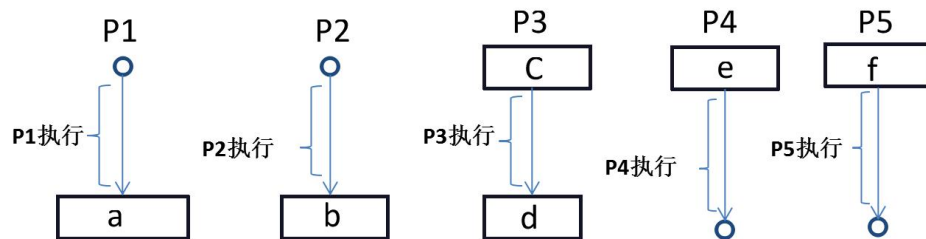
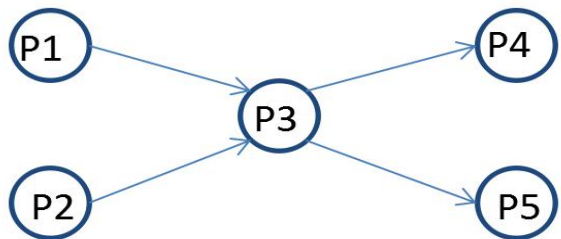
生产一个产品；
 $P(s1)$ ；
送产品到缓冲区；
 $V(s2)$ ；

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

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)

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。如果系统至少有多少个资源，则不可能发生死锁。

进程A	进程B	进程C

➡ 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，但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

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



还需资源数表

进程 \ 资源	最大需求量			已分配资源数			还需资源数		
	R1	R2	R3	R1	R2	R3	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

首先求剩下的资源数:

$$R1=9 (1+2+2+1+1) =2$$

$$R2=8 (2+1+1+2+1) =1$$

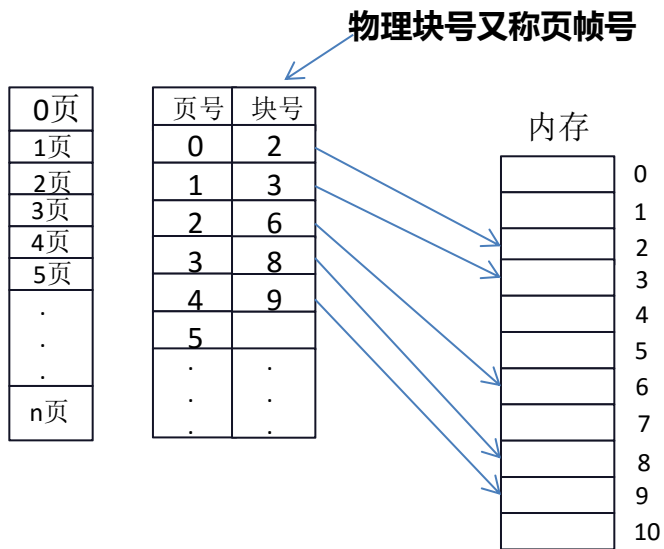
$$R3=5 (1+1+3) =0$$

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			需要资源			已经分配			现有+已经分配			完成
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
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

页式存储：将程序与内存均划分为同样大小的块，以页为单位将程序调入内存



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。
逻辑地址=页号+页内地址
物理地址=页帧号+页内地址

例如，页式存储系统中，每个页的大小为4KB。
逻辑地址：
10 1100 1101 1110
对应的物理地址为：
110 1100 1101 1110

★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

高级程序语言 中使用

内存中使用

1: 在内存中

0: 不在内存中

1: 最近访问过

0: 最近未被访问

1: 内容被修改过

0: 内容未被修改过

页号 (逻辑)	页帧号 (物理)	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

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

设某进程的段表如下所示，逻辑地址（ ）可以转换为对应的物理地址。

段号	基地址	段长
0	1598	600
1	486	50
2	90	100
3	1327	2988
4	1952	960

- A. (0 ,1597) 、 (1, 30) 和 (3, 1390)
- B. (0 ,128) 、 (1, 30) 和 (3 , 1390)
- C. (0 ,1597) 、 (2, 98) 和 (3, 1390)
- D. (0 ,128) 、 (2, 98) 和 (4, 1066)



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

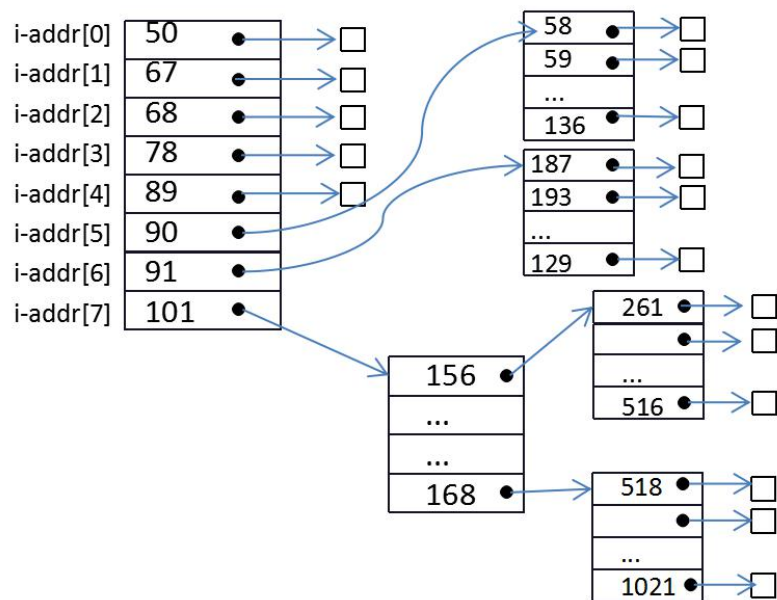
C. 58和187 D. 90和136

(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



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。

31	30	...		3	2	1	0
0	1	...	1	0	0	0	1

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”

第1字	1	1	1	0	0	...	1	1
第2字	0	1	1		0	...	0	1
第3字	1	1	1	1	0	...	1	0
...	...							
第n字	0	0	0	1	1	...	0	0

位示图例

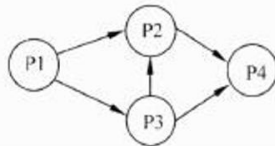
$(4195+1)/32=131.125 \rightarrow$ 第132字。

$131*32=4192 \rightarrow 0 - 4191.$

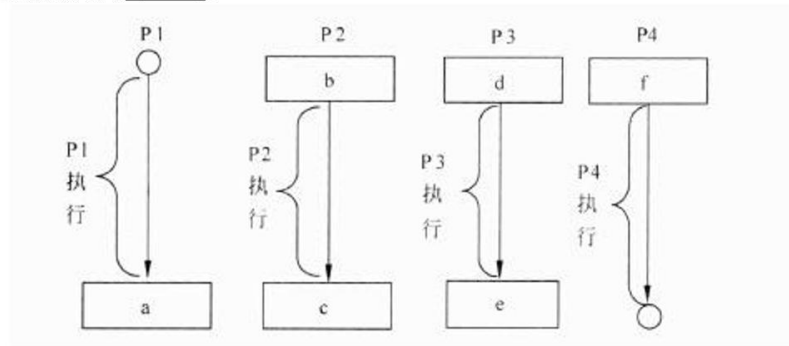
第132字中:

第0位置 $\rightarrow 4192$ 第1位置 $\rightarrow 4193$ 第2位置 $\rightarrow 4194$ 第3位置 $\rightarrow 4195$

- 进程 P1、P2、P3 和 P4 的前趋图如下所示：



若用 PV 操作控制进程 P1~P4 并发执行的过程，则需要设置 5 个信号量 S1、S2、S3、S4 和 S5，且信号量 S1~S5 的初值都等于 0。下图中 a、b 和 c 处应分别填写 (27) d、e 和 f 处应分别填写 (28)。



(27) A. V(S1) V(S2)、P(S1) V(S3) 和 V(S4)

C. V(S1) V(S2)、P(S1) P(S3) 和 V(S4)

(28) A. P(S2)、V(S3) V(S5) 和 P(S4) P(S5)

C. P(S2)、V(S3) P(S5) 和 P(S4) V(S5)

B. P(S1) V(S2)、P(S1) P(S2) 和 V(S1)

D. P(S1) P(S2)、V(S1) P(S3) 和 V(S2)

B. V(S2)、P(S3) V(S5) 和 V(S4) P(S5)

D. V(S2)、V(S3) P(S5) 和 P(S4) V(S5)

历年真题

答案CA, 解析:

