# 第一章：计算机组成与体系结构

## 流水线吞吐率、加速比

### 2017年下半年

1. 某计算机系统采用5级流水线结构执行指令，设每条指令的执行由取指令(2)、分析指令(1)、取操作数(3)、运算(1)和写回结果(2) 组成， 并分别用5个子部件完成，该流水线的最大吞吐率为( ) ；若连续向流水线输入10条指令，则该流水线的加速比为( ) 。

**【解析】**

理论流水线执行时间=(2+1+3+1+2)+max(2,1,3,1,2) \*(n-1) = 9+(n-1)\*3;

**第一问**：

最大吞吐率：

**第二问：**

10条指令使用流水线的执行时间=9+(10-1)\*3=36。

10条指令不用流水线的执行时间=9\*10=90。

加速比=使用流水线的执行时间/不使用流水线的执行时间=90/36 = 5:2。

### 其他

1. 例：某计算机系统，一条指令的执行需要经历取指（2ms）、分析（4ms）、执行（1ms）三个阶段，现要执行100 条指令，利用流水线技术需要多长时间? （教材1.3.1）

理论上来说，1条指令的执行时间为：2ms+4ms+1ms=7ms。

所以：理论流水线执行时间=2ms+4ms+1ms+(100-1)\*4=403ms。

而实际上，真正做流水线处理时，考虑到处理的复杂性，会将指令的每个执行阶段的时间都统一为流水线周期，即1条指令的执行时间为：4ms+4ms+4ms=12ms。所以：实际流水线执行时间=4ms+4ms+4ms+(100-1)\*4=408ms

**扩展：**

上述题目中，如果采用

3级操作，2级流水，等价于将3级操作变成2级操作。

最合理的划分是由取指（2ms）、分析（4ms）、执行（1ms）**相连**划分为指（2ms）、分析（4ms）+执行（1ms）={2,5}。

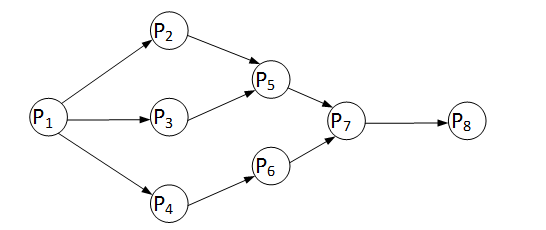
然后利用公式计算就是理论：(2+5)+(100-1)\*5=502，实际：(5+5)+(100-1)\*5=505。

# 第二章：操作系统

## PV操作、前趋图

### 2017年下半年

前趋图(Precedence Graph) 是一个有向无环图，记为：→={（Pi，Pj ）|Pi must complete before Pj may strat}。假设系统中进程P={P1，P2，P3，P4，P5，P6，P7，P8}，且进程的前驱图如下：



那么前驱图可记为：（ ）。

A: →={(P2,P1),(P3,P1),(P4,P1),(P6,P4),(P7,P5）,（P7,P6）,（P8,P7）}

B: →={（P1,P2）,(P1,P3),(P1,P4),(P2,P5),(P5,P7),(P6,P7）,（P7,P8）}

C: →={（P1,P2),(P1,P3),(P1,P4),(P2,P5),（P3,P5),(P4,P6),(P5,P7),(P6,P7),(P7,P8)}

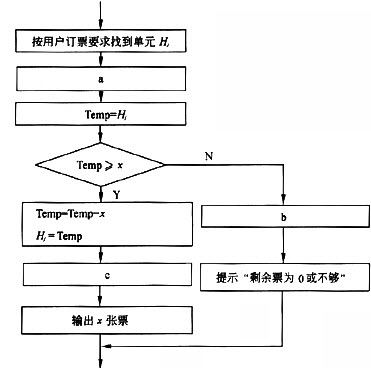
D: →={(P2, P1), (P3,P1),(P4,P1),(P5,P2),(P5,P2),(P5,P3),(P6,P4),(P7,P5), (P7,P6),(P8,P7)}

**【解析】**

容易得答案C。

### 2015年下半年

1. 某火车票销售系统有n个售票点，该系统为每个售票点创建一个进程Pi(i=1，2，…，n)。假设Hi(j=1，2+，…，m)单元存放某日某车次的剩余票数，Temp为Pi进程的临时工作单元，x为某用户的订票张数。初始化时系统应将信号量S赋值为 ( ) 。Pi进程的工作流程如下，若用P操作和V操作实现进程间的同步与互斥，则图中a、b和c应分别填入 ( ) 。



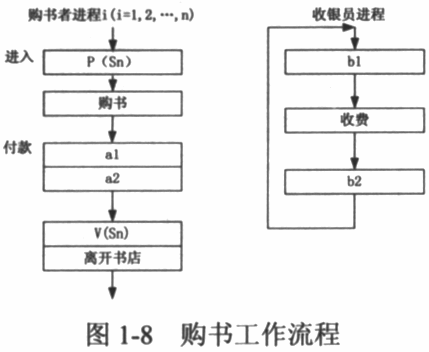
**【解析】**

第一空正确答案是1，因为公共数据单元马是一个临界资源，最多允许1个终端进程使用，因此需要设置一个互斥信号量S，初值等于1。

第二空的正确答案是P(S)、V(S)和V(S)，因为进入临界区时执行P操作，退出临界区时执行 V操作。（个人理解临界区就是菱形判断条件）。

### 2012年下半年

1. 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 ( ) ， b1和b2应填入 ( ) 。（2012年下半年）



**【解析】**

这是一道考查利用P、V操作实现进程间的同步工作的综合分析题。对于本试题收银员进程和购书者进程之间是一个同步问题，需要设置两个同步信号量，即S1和S2。其中，信号量 S1表示购书者购书时，通知收银员进程做收费工作，初值为0。信号量S2表示收银员收费结束，通知购书者进程可以进行一步工作，初值为0。

由于该书店最多只允许有n个购书者进入，因此，书店是一个临界资源，最多允许n个购书者购书，对应的是设置一个互斥信号量Sn，初值等于n。当购书者进入书店时需要执行P(Sn)操作，用于查看书店是否有空闲位置允许其进入购书。若有空闲位置，则进入书店进行购书；若没有空闲位置，则进入等待状态。当购书者完成购书操作退出书店时，需要执行V(Sn)操作，表明书店中已有一个空闲位置，并唤醒其他进入等待状态的购书者进程。

购书者进程中，完成购书操作后先执行V(S1)操作表示购书结束，唤醒收银员进程做收费工作。然后执行P(S2)，用于查看该购书者是否已缴费，若已缴费，则继续进行一步工作，即执行V(Sn)；若未缴费，则进入等待状态。

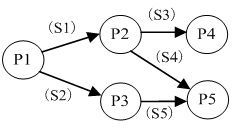
对于收银员进程，先执行P(S1)操作，用于检查是否有准备缴费的购书者申请。若有，则进行执行下一步工作，即进行收费操作；若没有准备缴费的购书者申请，则进入等待状态。当完成收费任务后，需继续执行V(S2)操作，用于通知购书者进程可以进行一步工作。

答案：V(S1)、P(S2)；P(S1)、V(S2)

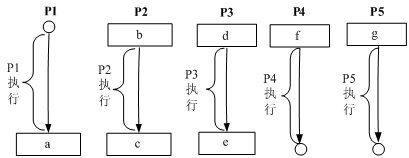
**可参考视频《0405。PV操作练习题1.wmv》解析。**

### 2011年下半年

1. 进程P1、P2、P3、P4和P5的前趋图如下：



       若用PV操作控制进程P1～P5并发执行的过程，则需要设置5个信号量S1、S2、S3、S4和S5，进程间同步所使用的信号量标注在上图中的边上，且信号量S1～S5的初值都等于零，初始状态下进程P1开始执行。下图中a、b和c处应分别填写（ ）；d和e处应分别填写（ ），f和g处应分别填写（ ）。



**【解析】**

最简单的理解方式：箭头出就是V操作，箭头入就是P操作。

答案：1、V(S1)V(S2)、P(S1)和V(S3)V(S4)；P(S2)和V(S5)；P(S3)和P(S4)P(S5)

### 其他

1. 有一个仓库可以存放P1、P2两种产品，但是每次只能存放一种产品。要求：（**不用太理解**）

①w=Num(P1)-Num(P2)；

②-i＜w＜k(i、k为正整数)。

如果Num(P1)=0，则-i<-Num(P2)<k，则-k<Num(P2)<i。所以仓库最多放i-1个P2产品；

若用P/V操作实现P1和P2产品的入库过程，则至少需要上 ( ) 个同步信号量及 ( ) 个互斥信号量。其中，同步信号量的初值分别为 ( ) ，互斥信号量的初值分别为 ( ) 。

**【解析】**

首先不看题，根据我的一般理解，一个系统中一般是问我有几个同步信号量，和互斥信号量，同步一般是2个，互斥一般是1个。同步的初值一般是0或者资源数，互斥的初值一般设为1

现在根据题目的第一句分析，一个仓库，放两种产品P1，P2，每次只能放一种。也就是说，有一个箱子，P1和P2都可以放，但是一次只能放P1，或者只能放P2，不能同时放，这和互斥很像，想想一下千军万马过独木桥，独木桥谁都能过，但是一次只能过一个。所以对于箱子而言是互斥的。

这里面有互斥，那么有没有同步。同步是指协作，谁和谁协作，没看出来，应该没有同步吧。初步答案，互斥1个，同步0个。

接着求初始值，看要求，w=Num(P1)-Num(P2)，而且-i<w<k，i和k还都是整数。原来P1和P2的产品量不是一个啊，是多个啊。还有数量限制。

w=Num(P1)-Num(P2)不明白。但是 -i<w=Num(P1)-Num(P2)<k。看选项结果初值应该和k，i都些些关系。如果假设一个极端，Num(P1)=0或者Num(P2)=0。

如果Num(P1)=0，则-i<-Num(P2)<k，则-k<Num(P2)<i。所以仓库最多放i-1个P2产品（**<i,i又为正数,只能是i-1**）； 而Num(P2)=0，则-i<Num(P1)<k，则仓库最多放k-1个P1产品。

然后仓库的操作过程可能是这样的。首先假设要放P1进入仓库，要看下仓库里是否有P2产品，如果有P1就不能放进去；如果没有P2产品，只有P1产品，还要看看P1产品的数量是否到达了i-1，如果已经到达了i-1，也不能放P1了。而对于产品P2，则要看是否有P1产品，是否数量达到了k-1。

尽管还是没有看出同步该有的协作，但是对于P1，P2产品应该用两个信号量S1，S2初始值分别为k-1和i-1，表示P1产品的数量，和P2产品的数量。每次放入一个P1产品，就P(S1)减少一个资源。P2同理。

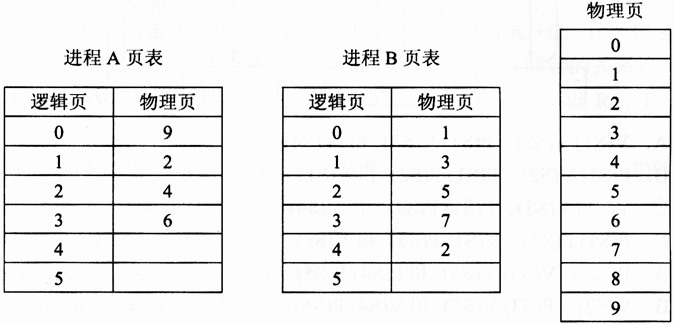
而互斥信号量，就是表示这个放了P1不能放P2，放了P2不能放P1。它的初值为1，放了P1之后，变为0，P2不能放，没资源了；或者放了P2之后变为0，P1不能放了。

所以同步为2，互斥为1；同步的初始值为i-1，k-1；互斥的初始值为1。

## 分页存储管理

### 2013下半年

1. 某操作系统采用分页存储管理方式，下图给出了进程A和进程B的页表结构。如果物理页的大小为512字节，那么进程A逻辑地址为1111(十进制)的变量存放在\_\_\_号物理内存页中。假设进程A的逻辑页4与进程B的逻辑页5要共享物理页8，那么应该在进程A页表的逻辑页4和进程B页表的逻辑页5对应的物理页处分别填\_\_\_。



**【解析】**

**第一问**：

十进制数1111转化为二进制数为：10001010111。物理页的大小为512字节，这说明页内地址为9个二进制位（2^9=512）。

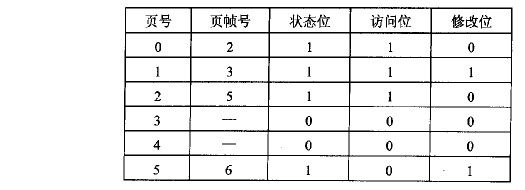
进程A的逻辑址中，右边的9位是页内地址，左边的2位是页号，即：10001010111。页号为二进制的10，即十进制的2，对应的物理页号为4。

**第二问**：

若A页表的逻辑页4和进程B页表的逻辑页5共享物理页8，则说明他们都对应物理页8，所以均填8（**物理页可以在进程间共享**）。

### 2012年下半年

1. 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1165H，那么该地址经过变换后，其物理地址应为十六进制\_\_\_：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\_\_\_的页面。



**【解析】**

**第一问**：

根据页式存储管理（页号查表+页内地址）

页面大小为4K 的二进制为2^12，则页内地址的位数为12位，高于12位的为页号。

二进制12位对应到十六进制的后三位（165H）。

再查表中页号1对应的物理块号（页帧号）为3，则物理地址为3165H。

**第二问**：

4不在内存，因为状态为为0，且题目告知。而页面的淘汰只能淘汰在内存中的。所以存页号0，、1、2、5中找一个淘汰，具体淘汰哪一个，就根据访问位确定；访问位位1的代表刚访问，不能淘汰，为0的才能淘汰，则淘汰5。

**对应《系统架构设计师考试全程指导》中19页的习题**。

1. 某操作系统采用分页存储管理方式，下图给出了进程A和进程B的页表结构。如果物理页的大小为512字节，那么进程A与进程B的物理内存总共使用了\_\_\_字节。

|  |  |  |  |
| --- | --- | --- | --- |
| 进程A页表： | | 进程B页表： | |
| 逻辑页 | 物理页 | 逻辑页 | 物理页 |
| 0 | 9 | 0 | 1 |
| 1 | 2 | 1 | 3 |
| 2 | 4 | 2 | 4 |
| 3 | 6 | 3 | 7 |
| 4 |  | 4 | 2 |
| 5 |  | 5 |  |

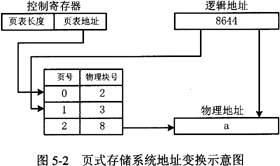
**【解析】**

物理页可以在进程间共享，两个进程共使用了1，2，3，4，6，7，9，共7个物理页。

故：7\*512=3584。

### 其他

1. 页式存储系统的逻辑地址是由页号和页内地址两部分组成。假定页面的大小为4KB，地址变换过程如图5-2所示。图5-2中有效地址经过变换后，十进制物理地址a应为（ ）。



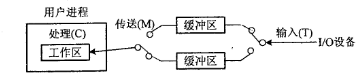
**【解析】**

因为页面大小为4KB（K），二进制为2^12，则页内地址的位数为12位，高于12位的为页号。

其中8644的二进制为1000111000100，则页号为10对应的十进制为2，物理块号为8（1000），重新组合成物理地址为100000111000100，将其转换为十进制为：33220。

## 文件系统

1. 某计算机系统输入/输出采用双缓冲工作方式，其工作过程如下图所示，假设磁盘块与缓冲区大小相同，每个盘块读入缓冲区的时间T为10，缓冲区送用户区的时间M为6，系统对每个磁盘块数据处理时间C为2。若用户需要将大小为10个磁盘块的Docl文件逐块从磁盘读入缓冲区，并送用户区进行处理，那么采用双缓冲需要花费的时间为（ ），比使用单缓冲节约了（ ）时间。



A．100 B．108 C．162 D．180

A．0 B．8 C．54 D．62

【解析】

**单缓冲区**：

假定从磁盘把一块数据输入到缓冲区的时间为T，操作系统将该缓冲区中的数据传送到用户区的时间为M，而CPU对这一块数据处理的时间为 C。

由于T和C是可以并行的，当T>C时，系统对每一块数据的处理时间为M+T，反之则为M+C，故可把系统对每一块数据的处理时间表示为max(C, T)+M。

单缓冲区执行时间：(10+6+2)+(10-1)\*(10+6)=162

**双缓冲区**：

系统处理一块数据的时间可以粗略地认为是max(C, T)。

双缓冲区执行时间：(10+6+2)+(10-1)\*10=108

双缓冲比单缓冲节省162-108=54。

1. 设文件索引结点中有8个地址项，每个地址项大小为4字节，其中5个地址项为直接地址索引，2个地址项是一级间接地址索引，1个地址项是二级间接地址索引，磁盘索引块和磁盘数据块大小均为1KB。则可表示的单个文件最大长度是多少KB？

【解析】

磁盘索引块为1KB字节，每个地址项大小为4字节，故每个磁盘索引块可存放1024/4=256个物理地址块。

又因为文件索引节点中有8个地址项，其中5个地址项为直接地址索引，这意味着逻辑块号为0—4的为直接地址索引。

2个地址项是一级间接地址索引，这意味着其中第一个地址项指出的物理块中存放逻辑块号为5—260的物理块号，其中第二个地址项指出的物理块中存放逻辑块号为261—516的物理块号。

1个地址项是二级间接地址索引，该地址项指出的物理块存放了256个间接索引表的地址，这256个间接索引表存放逻辑块号为517—66052的物理块号（256\*256=65536个）。

单个文件的逻辑块号范围是0—66052，而磁盘数据块大小为1KB，所以单个文件最大长度为：66053KB。

1. 某文件系统文件存储采用文件索引节点法。假设文件索引节点中有8个地址项iaddr[0]～iaddr[7]，每个地址项大小为4字节，其中地址项iaddr[0]～iaddr[5]为直接地址索引，iaddr[6]是一级间接地址索引，iaddr[7]是二级间接地址索引，磁盘索引块和磁盘数据块大小均为4KB。该文件系统可表示的单个文件最大长度是（ ）KB。若要访问iclsClient.dll文件的逻辑块号分别为6、520和1030，则系统应分别采用（ ）。

A．1030 B．65796 C．1049606 D．4198424

【解析】

**第一问**：

因为磁盘索引块和磁盘数据块大小均为4KB，每个地址项大小为4字节，所以每个磁盘索引块和磁盘数据块可存放4KB/4=**1024**个物理地址块。

计算直接地址索引，0-5存放6个物理块号，对应文件长度6\*4KB，对应逻辑块号0—5。

计算一级间接地址索引，1024\*4KB，对应逻辑块号5+1—1024+5=6—1029。

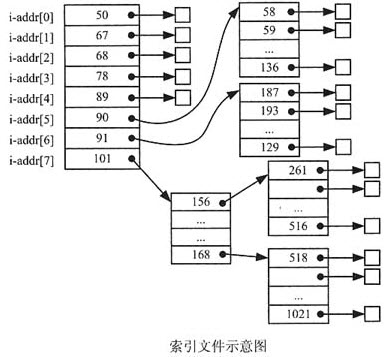
计算二级间接地址索引，1024\*1024\*4KB，对应逻辑块号1030及以上。

总计6\*4KB+1024\*4KB+1024\*1024\*4KB=4198424KB。

**第二问**：

由第一问对应的逻辑号，可得逻辑块号6、520和1030分别对应一级间接地址索引、一级间接地址索引、二级间接地址索引。

1. 假设文件系统采用索引节点管理，且索引节点有8个地址项iaddr[0]～iaddr[7]，每个地址项大小为4字节，iaddr[0]～iaddr[4]采用直接地址索引，iaddrl[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的索引节点如图所示。若用户访问文件Filel中逻辑块号为5和261的信息，则对应的物理块号分别为（ ）；101号物理块存放的是（ ）。



(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

【解析】

根据题意，磁盘索引块为1KB字节，每个地址项大小为4字节，故每个磁盘索引块可存放1024/4=256个物理块地址。又因为文件索引节点中有8个地址项，其中5个地址项为直接地址索引，这意味着逻辑块号为0～4的为直接地址索引；2个地址项是一级间接地址索引，其中第一个地址项指出的物理块中是一张一级间接地址索引表，存放逻辑块号为5～260对应的物理块号，第二个地址项指出的物理块中是另一张一级间接地址索引表，存放逻辑块号为261～516对应的物理块号。经上分析，从题图不难看出，逻辑块号为5的信息应该存放在58号物理块中，逻辑块号为261的信息应该存放在187号物理块中。

由题中可知，iaddr[7] 采用二级间接地址索引，且iaddr[7]中存放的物理块号为101，故101号物理块存放的是二级间接地址索引表。另外从示意图可以看出，101号物理块对应的空间存储着一系列地址，而这些地址对应的物理块中存储的仍然是地址，再到下一层才是文件内容，所以101号物理块存放的是二级地址索引表。

## 银行家算法

# 第三章：数据库系统

## 数据库关系模式、函数依赖

### 解题思路

##### 候选键（码）思路一

**闭包**：闭包就是由一个属性直接或间接推导出的所有属性的集合，记作**R+**。

**候选键（码）**：若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选键。在解题中的个人理解，候选键（码）能推导出U中所有元素（候选键（码）的闭包就是U），或者是消除冗余字段的[超键](https://blog.csdn.net/liushuijinger/article/details/12832017)。

假设：R<U,F>,U=(A,B,C,D,E,G),F={AB-→C,CD-→E,E-→A.A-→G},

1. 候选键（码）的计算：只出现在“→”左边B,D的一定是候选键（码），只出现在右边的G一定不是候选键（码）。
2. 然后根据选取/排除之后的数据进行组合，即BD可以跟A,C,E进行组合，因为BD不能直接推导出其他元素。
3. 先看ABD

ABD本身自包ABD,而AB→C,CD→E,A→G,所以ABD的闭包为ABDCEG=U

再看BDC

CD→E,E→A,A→G,BDC本身自包,所以BDC的闭包为BDCEAG=U

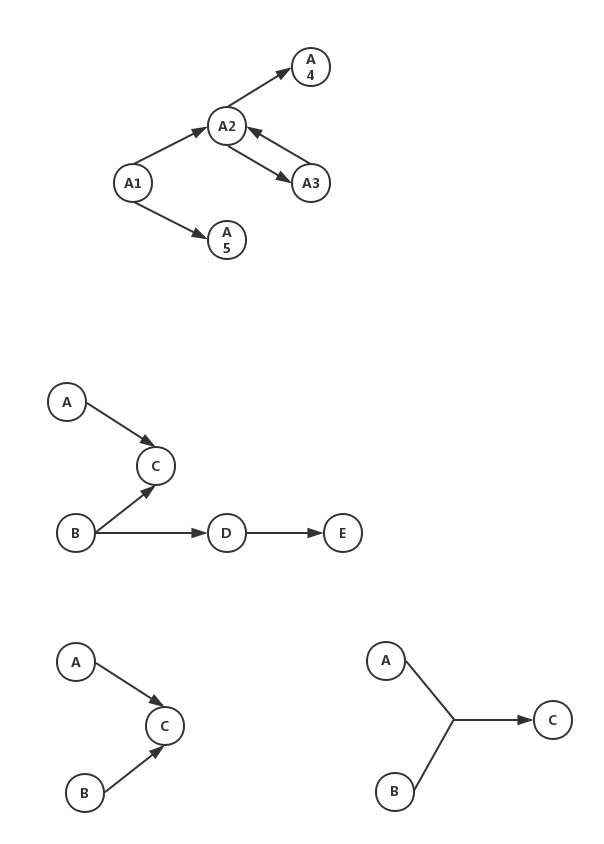
最后看BDE

E→A,A→G,AB→C,BDE本身自包,所以BDE的闭包为BDEAGC=U

1. 因为(ABD)、(BCD)、(BDE)的闭包都是ABCDEG所以本问题的候选码有3个分别是ABC、BCD和BDE

##### 候选键（码）思路二

1. 也通过绘制函数依赖图可以了解到，找到候选键（码），从每个元素出发，可以遍历全图（[例如2016年下半年例题](#_2016年下半年)），
2. **重点**：其中{AB→C}，只能画成右边的，不能画成左边的，因为左边代表A能确定C，B也能确定C。



### 2017年下半年

1. 给定关系模式R(U，F)，其中: 属性集 U={A1 ,A2,A3,A4,A5,A6}， 函数依赖集F={A1→A2, A1→A3, A3→A4, A1A5→A6}。关系模式 R 的候选码为（ ），由于R存在非主属性对码的部分函数依赖，所以R属于（1NF ）。

A: A1A3

B: A1A4

C: A1 A5

D: A1A6

**【解析】**

A1A5只出现在左边，是候选关键字。

### 2016年下半年

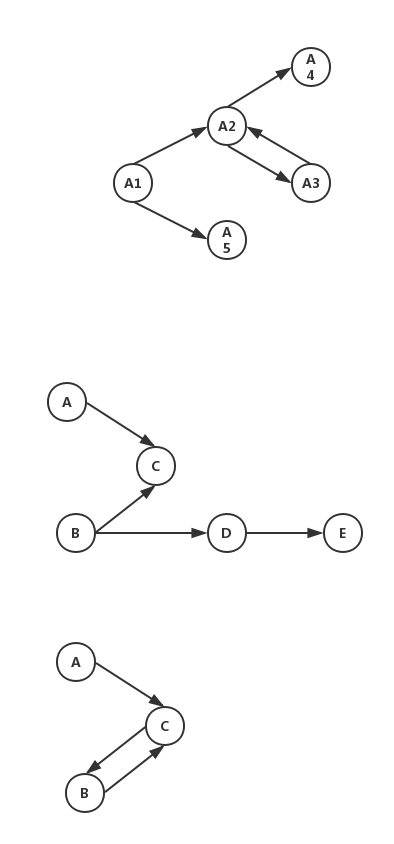
1. 给定关系R（A1，A2，A3，A4）上的函数依赖集F={A1→A2A5，A2→A3A4，A3→A2}，R的候选关键字为（ ）。函数依赖（ ）∈F+。

**【解析】**

**第一问：**

**解法一**：A1只出现在左边，是候选键；A4、A5只出现在右边，不是候选键。且A1的闭包等于R。所以A1为候选关键字。

**解法二**： 通过绘制函数依赖图可以了解到，从A1出发，可以遍历全图，所以候选关键字为A1。



**第二问：**

函数依赖（ ）∈F+，通俗一点，就是从F函数依赖集能推导出来的依赖关系。

A.A5→A1A2 B.A4→A1A2 C.A3→A2A4 D.A2→A1A5

根据函数依赖图可以看出C选项能走通。

### 其他

1. 假设关系模式R(U，F)，属性集U={A，B，C}，函数依赖集F={A→B，B→C}。若将其分解为ρ={R1(U1，F1)，R2(U2，F2)}，其中U1={A，B}，U2={A，C}。那么，关系模式R、R1、R2分别达到了\_\_\_；分解ρ\_\_\_。

**【解析】**

**第一问：**

由关系模式R的函数依赖集F={A→B，B→C}可以得出A→C，存在传递依赖，但不存在非主属性对码的部分函数依赖（**下面例题【解析】中的b)**），故R为2NF。

又由于分解后的关系模式R1的函数依赖集F1={A→B}，关系模式R2的函数依赖集F2={A→C}，因此R1、R2分别达到了3NF。

**第二问：**

表示能通过F推导出R1∩R2={A}，R1 - R2={B}， R2 - R1={C}。

R1∩R2→R1 - R2，为无损分解（不需要继续判断R1∩R2→R2 - R1，且判断结果为真）且不保持函数依赖（而R2中的A与C两个属性，没有保持任何函数依赖，导致函数依赖B→C丢失，所以分解没有保持函数依赖）。

1. 设关系模式R（A,B,C,D,E）,其函数依赖F={AB→C,B→D,D→E },完成下述各题.

a) 求出R的所有候选键；

b) 试分析关系R属于何种范式；

c) 将R分解为满足3NF的关系；

**【解析】**

a)R的候选键是AB,因为AB+=（ABCDE）,没有其他候选键了.  
 b)R只能是第一范式,因为B→D,存在非关键字部分依赖于候选键（“→”左边只有AB中的B）,所以不符合第2范式的条件.  
 c)第3范式就是在第2范式的基础上,不存在非关键字对任一候选键的传递依赖.所以把范式分解到符合第3范式的要求就可以了，R1{A,B,C},R2{B,D},R3{D,E}

## 无损分解

1. 设有关系模式R(U，V，W，X，Y，Z)，其函数依赖集：F＝{U→V，W→z，Y→U，WY→X}，现有下列分解：p＝{UVY，WXYZ}

**【解析】**

**判断分解p是否为无损连接**：若关系模式R(U,F)中，被分解为p={R1, R2}是R的一个分解，若R1∩R2 → R1 - R2或者R1∩R2 → R2 - R1，则为无损连接，**此方法只适用于分解后的关系模式只有两个**。

备注：表示能通过F推导出R1∩R2 → R1 - R2或者R1∩R2 → R2 - R1等关系。

根据判断标准，可得R1∩R2=Y，R1-R2=UV，能否通过F推导出Y →UV？

因为F中Y→U、U→V，可得Y →UV，即为无损连接。

不需要继续判断Y→R2 - R1。

1. 多个关系模式（有点复杂，例题可参考）

<https://wenku.baidu.com/view/6e435998bceb19e8b8f6bab1.html>

## 最小函数集

1. 求F={abd→e,ab→g,b→f,c→j,**cj→i**,g→h}的最小函数集。

**步骤一**：将F中的所有依赖右边化为单一元素，假如F中有cj→ik，则F单一化变成：

F={abd→e,ab→g,b→f,c→j,**cj→i,cj→k**,g→h}

但是，此题F已经满足此要求，不需要变换。

**步骤二**：去掉F中的所有依赖左边的冗余属性（“→”左边有多个元素）

做法是属性中去掉其中的一个，看看是否依然可以推导？

此题:abd→e，去掉a，则(bd)+不含e，故不能去掉，同理b,d都不是冗余属性。

ab→g，也没有。

cj→i，去掉j，c+={c,j,i}【是因为c能推导本身，c又能推导出j，cj又能推导出i)】 其中包含i所以j是冗余的。cj→i将成为c→i。

此时：F={abd→e,ab→g,b→f,c→j,c→i,g→h};

**步骤三**：去掉F中所有冗余依赖关系.

做法为从F中去掉某关系，如去掉(X→Y)，然后在F中求X+，如果Y在X+中,则表明X→是多余的，需要去掉。

此题如果F去掉abd→e，F将等于{ab→g,b→f,c→j,c→i,g→h}，而(abd)+={a,b,d,f,g,h}，其中不包含e。所有不是多余的。

同理(ab)+={a,b,f}也不包含g，故不是多余的。

b+={b}不多余，c+={c,i}不多余。

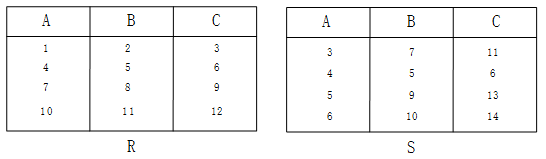
c→i，g→h都不能去掉。

所以所求最小函数依赖集为 F={abd→e,ab→g,b→f,c→j,c→i,g→h}。

## 元组演算表达式

**重点：1、其中元组表示关系表的行，t[4]表示元组 t 的第 4 个分量 。**

**2、选择运算****表示：选取R中行的第3个属性等于第6个属性元组，**

1. 给定元组演算表达式R\*={t│(**Э**u)(R(t)∧S(u)∧t[3]<u[2])} ，若关系 R、S如下图所示，则（  ）。  
     
    A.R\*={(3,7,11),(5,9,13),(6,10,14)}  
    B.R\*={(3,7,11),(4,5,6),(5,9,13),(6,10,14)}  
    C.R\*={(1,2,3),(4,5,6),(7,8,9)}  
    D.R\*={(1,2,3),(4,5,6),(7,8,9),(10,11,12)}

**【解析】**

题目中表达式：存在从关系R中选择的元组t的C列上的分量，大于关系S中的一个元组u在B列上的分量。

t[3]<u[2]：R中每行的第三个分量（R的第3列）<S中每行的第二个分量

t[3]={3,6,9,12}，u[2]={7,5,9,10}

t[3]中的3<{7,5,9,10}中的7,5,9,10，满足要求。

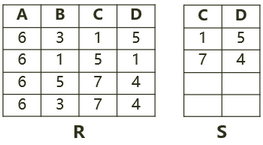
t[3]中的6<{7,5,9,10}中的7,9,10，满足要求。

t[3]中的9<{7,5,9,10}中的10，满足要求。

t[3]中的12不满足要求。**存在：只要满足u[2]中一个分量就行**。

所以t[3]<u[2] = {(1,2,3),(4,5,6),(7,8,9)}

1. 若关系R、S如下图所示，则关系R与S进行自然连接运算后的元组个数和属性列数分别为（）；关系代数表达式与关系代数表达式（）等价。



1. 3和4
2. 

**【第一问解析】**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | D |
| 6 | 3 | 1 | 5 |
| 6 | 5 | 7 | 4 |
| 6 | 3 | 7 | 4 |

**【第二问解析】**：(解析、计算结果)

其中结果为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | C | D |
| 6 | 3 | 1 | 5 | 1 | 5 |
| 6 | 3 | 1 | 5 | 7 | 4 |
| 6 | 1 | 5 | 1 | 1 | 5 |
| 6 | 1 | 5 | 1 | 7 | 4 |
| 6 | 5 | 7 | 4 | 1 | 5 |
| 6 | 5 | 7 | 4 | 7 | 4 |
| 6 | 3 | 7 | 4 | 1 | 5 |
| 6 | 3 | 7 | 4 | 7 | 4 |

：上述结果中行的第3个属性等于第6个属性，即R.C=S.D，结果为：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | C | D |
| 6 | 1 | 5 | 1 | 1 | 5 |

：对上述结果投影第1个和第4个属性列，即R中的A（记作A：只有R中有A）、R中的D（记作R.D），结果为

|  |  |
| --- | --- |
| A | D |
| 6 | 1 |

# 第十二章嵌入式系统

1. 内存按字节编址，地址从90000H到CFFFFH，若用存储容量为16K×8bit器芯片构成该内存,至少需要的存储 （ ）片。

**【解析】**

（CFFFFH-90000H+1）=3FFFFH+1=40000H

这是计算地址从90000H到CFFFFH的字节总容量，+1是因为要包含90000H该地址。

**方法一**：十六进制转十进制

40000H化为十进制为256K。由于内存是按照字节编址（默认8bit），所以存储容量：

(256K×8bit)/(16K×8bit)=16

**方法二**：十进制转十六进制

16K转成十六进制4000H，40000H/4000H=10，转成十进制为16。