

# 参考答案

## 一、单项选择题（后附有解析）

1. D      2. C      3. D      4. C      5. B      6. A      7. C      8. B  
9. B      10. D      11. A      12. D      13. B      14. B      15. D      16. A  
17. D      18. B      19. A      20. D      21. A      22. D      23. A      24. C  
25. B      26. A      27. D      28. B      29. B      30. C      31. C      32. B  
33. C      34. C      35. D      36. C      37. B      38. D      39. A      40. A

## 二、综合应用题

### 41. 解答：

(1) 由装载因子 0.7，数据总数为 7，得一维数组大小为  $7/0.7=10$ ，数组下标为 0~9。所构造的散列函数值如下所示：

|        |   |   |    |    |    |   |    |
|--------|---|---|----|----|----|---|----|
| key    | 7 | 8 | 30 | 11 | 18 | 9 | 14 |
| H(key) | 0 | 3 | 6  | 5  | 5  | 6 | 0  |

采用线性探测再散列法处理冲突，所构造的散列表为：

|     |   |    |   |   |   |    |    |    |   |   |
|-----|---|----|---|---|---|----|----|----|---|---|
| 地址  | 0 | 1  | 2 | 3 | 4 | 5  | 6  | 7  | 8 | 9 |
| 关键字 | 7 | 14 |   | 8 |   | 11 | 30 | 18 | 9 |   |

(2) 查找成功时，是根据每个元素查找次数来计算平均长度，在等概率的情况下，各关键字的查找次数为：

|     |   |   |    |    |    |   |    |
|-----|---|---|----|----|----|---|----|
| key | 7 | 8 | 30 | 11 | 18 | 9 | 14 |
| 次数  | 1 | 1 | 1  | 1  | 3  | 3 | 2  |

故， $ASL_{成功} = \text{查找次数} / \text{元素个数} = (1+2+1+1+1+3+3) / 7 = 12/7$

这里要特别防止惯性思维。查找失败时，是根据查找失败位置计算平均次数，根据散列函数 MOD 7，初始只可能在 0~6 的位置。等概率情况下，查找 0~6 位置查找失败的查找次数为：

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| H(key) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 次数     | 3 | 2 | 1 | 2 | 1 | 5 | 4 |

故， $ASL_{不成功} = \text{查找次数} / \text{散列后的地址个数} = (3+2+1+2+1+5+4) / 7 = 18/7$

### 42. 解答：

(1) 算法的基本设计思想：

可以将这个问题看做是把数组 ab 转换成数组 ba（a 代表数组的前 p 个元素，b 代表数组中余下的 n-p 个元素），先将 a 逆置得到  $a^{-1}b$ ，再将 b 逆置得到  $a^{-1}b^{-1}$ ，最后将整个  $a^{-1}b^{-1}$  逆置得到  $(a^{-1}b^{-1})^{-1}=ba$ 。设 Reverse 函数执行将数组元素逆置的操作，对 abcdefgh 向左循环移动 3（p=3）个位置的过程如下：

Reverse(0,p-1)得到 cbadefgh；

Reverse(p,n-1)得到 cbahgfed；

Reverse(0,n-1)得到 defghabc；

注：Reverse 中，两个参数分别表示数组中待转换元素的始末位置。

(2) 使用 c 语言描述算法如下：

```
void Reverse(int R[],int from,int to) {  
    int i,temp;  
    for(i = 0; i < (to-from+1)/2; i++)
```

```

    {    temp = R[from+i]; R[from+i] = R[to-i]; R[to-i] = temp; }
} // Reverse

void Converse(int R[],int n,int p){
    Reverse(R,0,p-1);
    Reverse(R,p,n-1);
    Reverse(R,0,n-1);
}

```

(3) 上述算法中三个 Reverse 函数的时间复杂度分别为  $O(p/2)$ 、 $O((n-p)/2)$  和  $O(n/2)$ ，故所设计的算法的时间复杂度为  $O(n)$ ，空间复杂度为  $O(1)$ 。

**另解**，借助辅助数组来实现：

算法思想：创建大小为  $p$  的辅助数组  $S$ ，将  $R$  中前  $p$  个整数依次暂存在  $S$  中，同时将  $R$  中后  $n-p$  个整数左移，然后将  $S$  中暂存的  $p$  个数依次放回到  $R$  中的后续单元。

时间复杂度为  $O(n)$ ，空间复杂度为  $O(p)$ 。

#### 43. 解答：

(1) 操作码占 4 位，则该指令系统最多可有  $2^4=16$  条指令；操作数占 6 位，寻址方式占 3 位，于是寄存器编号占 3 位，则该机最多有  $2^3=8$  个通用寄存器；主存容量 128KB，按字编址，计算机字长为 16 位，划分为  $128KB/2B=2^{16}$  个存储单元，故 MDR 和 MAR 至少各需 16 位。

(2) PC 和  $R_n$  可表示的地址范围均为  $0 \sim 2^{16}-1$ ，而主存地址空间为  $2^{16}$ ，故转移指令的目标地址范围是 0000H~FFFFH ( $0 \sim 2^{16}-1$ )。

(3) 汇编语句 “add (R4), (R5)+”，对应的机器码为 0010 0011 0001 0101B=2315H。

该指令执行后，寄存器 R5 和存储单元 5678H 的内容会改变。执行后，R5 的内容从 5678H 变成 5679H。存储单元 5678H 中的内容变成该加法指令计算的结果  $5678H+1234H=68ACH$ 。

#### 44. 解答：

(1) 数据 Cache 有 8 个 Cache 行，每个 Cache 行大小为 64B，Cache 中每个字块的 Tag 字段的位数是  $28-9=19$  位，此外还需使用一个有效位，合计 20 位。因此，数据 Cache 的总容量应为： $8 \times (64 + 20 / 8)B = 532B$ 。

(2) 数组 a 在主存的存放位置及其与 Cache 之间的映射关系如下图所示：

数组按行优先方式存放，首地址为 320，数组元素占四个字节。a[0][31]所在的主存块对应的 Cache 行号为：

$$(320 + 31 \times 4) \text{ DIV } 64 = 6 ;$$

a[1][1]所在的主存块对应的 Cache 行号为：

$$(320 + 256 \times 4 + 1 \times 4) \text{ DIV } 64 \text{ MOD } 8 = 5 .$$

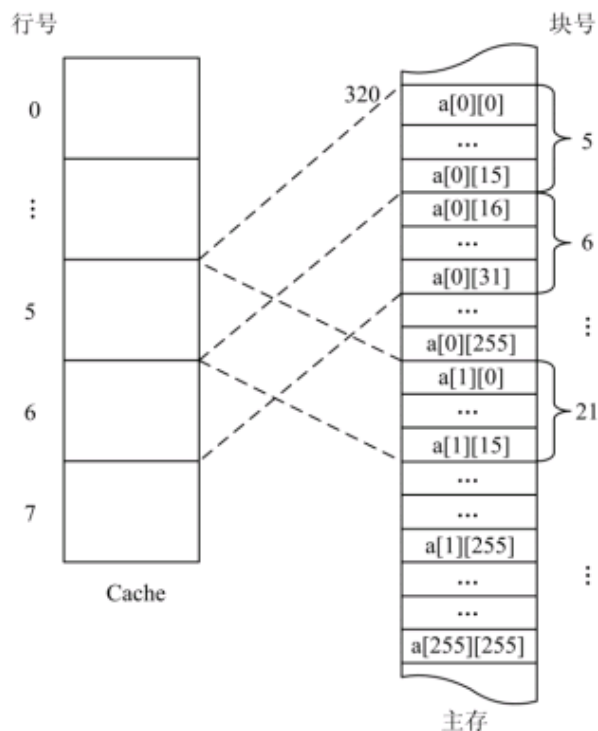
(3) 编译时 i、j、sum 均分配在寄存器中，故数据访问命中率仅考虑数组 a 的情况。

① 该程序的特点是数组中的每一个元素仅被使用一次。数组 a 按行优先存放，数据 Cache 正好放下数组半行中的全部元素，即元素的存储顺序与使用次序高度的吻合，每个字块的 16 个 int 型元素中，除访问的第一个不会命中，接下来的 15 个都会命中。访问全部字块都符合这一规律，故命中率为  $15 / 16$ ，即程序 A 的数据访问命中率是 93.75%。

② 程序 B 按照数组的列执行外层循环，在执行内层循环的过程中，将连续访问不同行的同一列的数据，不同行的同一列数组使用的是同一个 Cache 单元，每次都不会命中，故命中率是 0

由于从 Cache 读数据比从主存读数据快很多，所以程序 A 的执行比程序 B 快得多。

**注意：** 本题考查 Cache 容量计算，直接映射方式的地址计算，以及命中率计算（注意：行优先遍历与列优先遍历命中率差别很大）。



45. 解答:

(1) 用位图表示磁盘的空闲状态。每一位表示一个磁盘块的空闲状态,共需要  $16384 / 32 = 512$  个字  $= 512 \times 4$  个字节  $= 2\text{KB}$ ,正好可放在系统提供的内存中。

(2) 采用 CSCAN 调度算法,访问磁道的顺序和移动的磁道数如下表所示:

| 被访问的下一个磁道号 | 移动距离 (磁道数) |
|------------|------------|
| 120        | 20         |
| 30         | 90         |
| 50         | 20         |
| 90         | 40         |

移动的磁道数为  $20 + 90 + 20 + 40 = 170$ ,故总的移动磁道时间为  $170\text{ms}$ 。

由于转速为  $6000\text{r/m}$ ,则平均旋转延迟为  $5\text{ms}$ ,总的旋转延迟时间  $= 20\text{ms}$ 。

由于转速为  $6000\text{r/m}$ ,则读取一个磁道上一个扇区的平均读取时间为  $0.1\text{ms}$ ,总的读取扇区的时间平均读取时间为  $0.1\text{ms}$ ,总的读取扇区的时间为  $0.4\text{ms}$ 。

综上,读取上述磁道上所有扇区所花的总时间为  $190.4\text{ms}$ 。

(3) 采用 FCFS (先来先服务) 调度策略更高效。因为 Flash 半导体存储器的物理结构不需要考虑寻道时间和旋转延迟,可直接按 I/O 请求的先后顺序服务。

46. 解答:

(1) 由于该计算机的逻辑地址空间和物理地址空间均为  $64\text{KB} = 2^{16}\text{B}$ ,按字节编址,且页的大小为  $1\text{K} = 2^{10}$ ,故逻辑地址和物理地址的地址格式均为:

|             |             |
|-------------|-------------|
| 页号/页框号 (6位) | 页内偏移量 (10位) |
|-------------|-------------|

$17\text{CAH} = 0001\ 0111\ 1100\ 1010\text{B}$ ,可知该逻辑地址的页号为  $000101\text{B} = 5$

(2) 根据 FIFO 算法,需要替换装入时间最早的页,故需要置换装入时间最早的 0 号页,即将 5 号页装入 7 号页框中,所以物理地址为  $0001\ 1111\ 1100\ 1010\text{B} = 1\text{FCAH}$ 。

(3) 根据 CLOCK 算法,如果当前指针所指页框的使用位为 0,则替换该页;否则将使用位清零,并将指针指向下一个页框,继续查找。根据题设和示意图,将从 2 号页框开始,前 4 次查找页框号的顺序为

---

2→4→7→9，并将对应页框的使用位清零。在第5次查找中，指针指向2号页框，因2号页框的使用位为0，故淘汰2号页框对应的2号页，把5号页装入2号页框中，并将对应使用位设置为1，所以对应的物理地址为0000 1011 1100 1010B = 0BCAH。

47. 解答：

(1) 当主机甲和主机乙同时向对方发送数据时，信号在信道中发生冲突后，冲突信号继续向两个方向传播。这种情况下两台主机均检测到冲突需要经过的时间最短，等于单程的传播时延  $t_0 = 2\text{km}/200\,000\text{km/s} = 0.01\text{ms}$ 。

主机甲（或主机乙）先发送一个数据帧，当该数据帧即将到达主机乙（或主机甲）时，主机乙（或主机甲）也开始发送一个数据帧，这时，主机乙（或主机甲）将立刻检测到冲突，而主机甲（或主机乙）要检测到冲突，冲突信号还需要从主机乙（或主机甲）传播到主机甲（或主机乙），因此甲乙两台主机均检测到冲突所需的最长时间等于双程的传播时延  $2 \cdot t_0 = 0.02\text{ms}$ 。

(2) 主机甲发送一个数据帧的时间，即发送时延  $t_1 = 1518 \times 8\text{b}/10\text{Mbps} = 1.2144\text{ms}$ ；主机乙每成功收到一个数据帧后，向主机甲发送确认帧，确认帧的发送时延  $t_2 = 64 \times 8\text{b}/10\text{Mbps} = 0.0512\text{ms}$ ；主机甲收到确认帧后，即发送下一数据帧，故主机甲的发送周期  $T = \text{数据帧发送时延 } t_1 + \text{确认帧发送时延 } t_2 + \text{双程传播时延} = t_1 + t_2 + 2 \cdot t_0 = 1.2856\text{ms}$ ；于是主机甲的有效数据传输率为

$1500 \times 8 / T = 12000\text{b} / 1.2856\text{ms} \approx 9.33\text{Mbps}$ 。（以太网有效数据1500字节，即以太网帧的数据部分）

**2010 年全国硕士研究生入学统一考试**  
**计算机学科专业基础综合试题——选择题解析**

**一、单项选择题**

1. 考查限定条件的出栈序列。

A 可由 in, in, in, in, out, out, in, out, out, in, out, out 得到;

B 可由 in, in, in, out, out, in, out, out, in, out, in, out 得到;

C 可由 in, in, out, in, out, out, in, in, out, in, out, out 得到;

D 可由 in, out, in, in, in, in, in, out, out, out, out, out 得到, 但题意要求不允许连续三次退栈操作, 故 D 错。

2. 考查受限的双端队列的出队序列。

A 可由左入, 左入, 右入, 右入, 右入得到 B 可由 左入, 左入, 右入, 左入, 右入得到。

D 可由左入, 左入, 左入, 右入, 左入得到 所以不可能得到 C。

3. 考查线索二叉树的基本概念和构造。

题中所给二叉树的后序序列为 dbca。结点 d 无前驱和左子树, 左链域空, 无右子树, 右链域指向其后继结点 b; 结点 b 无左子树, 左链域指向其前驱结点 d; 结点 c 无左子树, 左链域指向其前驱结点 b, 无右子树, 右链域指向其后继结点 a。

4. 考查平衡二叉树的插入算法。

插入 48 以后, 该二叉树根结点的平衡因子由-1 变为-2, 失去平衡, 需进行两次旋转(先右旋后左旋)操作。

5. 考查树结点数的特性。

设树中度为  $i$  ( $i=0, 1, 2, 3, 4$ ) 的结点数分别为  $N_i$ , 树中结点总数为  $N$ , 则树中各结点的度之和等于  $N-1$ , 即  $N = 1 + N_1 + 2N_2 + 3N_3 + 4N_4 = N_0 + N_1 + N_2 + N_3 + N_4$ , 根据题设中的数据, 即可得到  $N_0 = 82$ , 即树 T 的叶结点的个数是 82。

6. 考查哈弗曼树的特性。

哈弗曼树为带权路径长度最小的二叉树, 不一定是完全二叉树。哈弗曼树中没有度为 1 的结点, B 正确; 构造哈弗曼树时, 最先选取两个权值最小的结点作为左右子树构造一棵新的二叉树, C 正确; 哈弗曼树中任一非叶结点 P 的权值为其左右子树根结点权值之和, 其权值不小于其左右子树根结点的权值, 在与结点 P 的左右子树根结点处于同一层的结点中, 若存在权值大于结点 P 权值的结点 Q, 那么结点 Q 的兄弟结点中权值较小的一个应该与结点 P 作为左右子树构造新的二叉树, 综上可知, 哈弗曼树中任一非叶结点的权值一定不小于下一层任一结点的权值。

7. 考查图的连通性。

要保证无向图 G 在任何情况下都是连通的, 即任意变动图 G 中的边, G 始终保持连通, 首先需要 G 的任意六个结点构成完全连通子图 G1, 需 15 条边, 然后再添一条边将第 7 个结点与 G1 连接起来, 共需 16 条边。

8. 考查拓扑排序序列。

题中图有三个不同的拓扑排序序列, 分别为 abced, abecd, aebcd。

9. 考查折半查找的过程。

具有  $n$  个结点的判定树的高度为  $\lfloor \log_2 n \rfloor + 1$ , 长度为 16, 高度为 5, 所以最多比较 5 次。

10. 考查快速排序。

递归次数与各元素的初始排列有关。如果每一次划分后分区比较平衡，则递归次数少，如果划分后分区不平衡，则递归次数多。递归次数与处理顺序无关。

**11. 考查各种排序算法的过程。**

看第一趟可知仅有 88 被移到最后。

如果是希尔排序，则 12, 88, 10 应变为 10, 12, 88。因此排除希尔排序。

如果是归并排序，则应长度为 2 的子序列是有序的，由此可排除归并。

如果是基数排序，则 16, 5, 10 应变为 10, 5, 16，由此排除基数。

可以看到，每一趟都有一个元素移到其最终位置，符合冒泡排序特点。

**12. 考查计算机的性能指标。**

I.CPU 的时钟频率，也就是 CPU 主频率，一般说来，一个时钟周期内完成的指令数是固定的，所以主频越高，CPU 的速度也就快，程序的执行时间就越短。

II.数据在功能部件之间传送的路径称为数据通路，数据通路的功能是实现 CPU 内部的运算器和寄存器以及寄存器之间的数据交换。优化数据通路结构，可以有效提高计算机系统的吞吐量，从而加快程序的执行。

III.计算机程序需要先转化成机器指令序列才能最终得到执行，通过对程序进行编译优化可以得到更优的指令序列，从而使得程序的执行时间也越短。

**13. 考查定点数的运算。**

用补码表示时 8 位寄存器所能表示的整数范围为-128~+127。由于  $r1 = -2$ ,  $r2 = -14$ ,  $r3 = -112$ ,  $r4 = -8$ , 则  $r2 \times r3 = 1568$ , 结果溢出。

**14. 考查不同精度的数在计算机中的表示方法及其相互装换。**

由于 (int) f=1, 小数点后面 4 位丢失, 故 II 错。IV 的计算过程是先将 f 转化为双精度浮点数据格式, 然后进行加法运算, 故 (d+f) -d 得到的结果为双精度浮点数据格式, 而 f 为单精度浮点数据格式, 故 IV 错。

**15. 考查存储器的组成和设计。**

用 2K×4 位的芯片组成一个 8K×8 位存储器, 每行中所需芯片数为 2, 每列中所需芯片数为 4, 各行芯片的地址分配为:

第一行 (2个芯片并联) 0000H~07FFH

第二行 (2个芯片并联) 0800H~0FFFFH

第三行 (2个芯片并联) 1000H~17FFH

第四行 (2个芯片并联) 1800H~1FFFFH

于是地址 0B1FH 所在芯片的最小地址即为 0800H。

**16. 考查半导体随机存取存储器。**

一般 Cache 采用高速的 SRAM 制作, 比 ROM 速度快很多, 因此 III 是错误的, 排除法即可选 A。

RAM 需要刷新, 而 ROM 不需要刷新。

**17. 考查 TLB、Cache 及 Page 之间的关系。**

TLB 即为快表, 快表只是慢表 (Page) 的小小副本, 因此 TLB 命中, 必然 Page 也命中, 而当 Page 命中, TLB 则未必命中, 故 D 不可能发生; 而 Cache 的命中与否与 TLB、Page 的命中与否并无必然联系。

**18. 考查 CPU 内部寄存器的特性。**

汇编程序员可以通过指定待执行指令的地址来设置 PC 的值, 而 IR, MAR, MDR 是 CPU 的内部工作寄存器, 对程序员不可见。

#### 19. 考查指令流水线的基本概念。

有三种相关可能引起指令流水线阻塞：1. 结构相关，又称资源相关；2. 数据相关；3. 控制相关，主要由转移指令引起。

数据旁路技术，其主要思想是不必待某条指令的执行结果送回到寄存器，再从寄存器中取出该结果，作为下一条指令的源操作数，而是直接将执行结果送到其他指令所需要的地方，这样可以使流水线不发生停顿。

#### 20. 考查典型的总线标准，

目前典型的总线标准有：ISA、EISA、VESA、PCI、PCI-Express、AGP、USB、RS-232C 等。

#### 21. 考查中断处理过程。

单级中断系统中，不允许中断嵌套。中断的处理过程为：1. 关中断；2. 保存断点；3. 识别中断源；4. 保存现场；5. 中断事件处理；（开中断、执行中断服务程序、关中断）6. 恢复现场；7. 开中断；8. 中断返回。其中，1~3 步由硬件完成，4~8 由中断服务程序完成。

#### 22. 考查显示器相关概念。

刷新所需带宽 = 分辨率×色深×帧频 =  $1600 \times 1200 \times 24 \times 85 \text{HZ} = 3916.8 \text{Mbps}$ ，显存总带宽的 50% 用来刷屏，于是需要的显存总带宽为  $3916.8 / 0.5 = 7833.6 \text{Mbps} \approx 7834 \text{Mbps}$ 。

#### 23. 考查操作系统的接口。

系统调用是能完成特定功能的子程序，当应用程序要求操作系统提供某种服务时，便调用具有相应功能的系统调用。库函数则是高级语言中提供的与系统调用对应的函数（也有些库函数与系统调用无关），目的是隐藏访管指令的细节，使系统调用更为方便抽象。但要注意，库函数属于用户程序而非系统调用，是系统调用的上层。

#### 24. 考查引起创建进程的事件。

引起进程创建的事件有：用户登录、作业调度、提供服务、应用请求等，本题的选项分别对应：I 用户登录成功 在分时系统中，用户登录成功，系统将为终端建立一个进程。II 设备分配 设备分配是通过在系统中设置相应的数据结构实现的，不需要创建进程。III 启动程序执行 典型的引起创建进程的事件。

#### 25. 考查信号量的原理。

信号量表示当前的可用相关资源数。当信号量  $K > 0$  时，表示还有  $K$  个相关资源可用；而当信号量  $K < 0$  时，表示有  $|K|$  个进程在等待该资源。所以该资源可用数是 1，等待该资源的进程数是 0。

#### 26. 考查进程调度。

进程时间片用完，从执行态进入就绪态应降低优先级以让别的进程被调度进入执行状态。B 中进程刚完成 I/O，进入就绪队列后应该等待被处理机调度，故应提高优先权；C 中有类似的情况；D 中不应该在此时降低，应该在时间片用完时降低。

#### 27. 考查进程间通信与 Peterson 算法。

此算法实现互斥的主要思想在于设置了一个 `turn` 变量，用于进程间的互相“谦让”。

一般情况下，如果进程 `p0` 试图访问临界资源，设置 `flag[0]=true`，表示希望访问。此时如果进程 `p1` 还未试图访问临界资源，则 `flag[1]` 在进程上一次访问完临界资源退出临界区后已设置为 `false`。所以进程 `p0` 在执行循环判断条件时，第一个条件不满足，进程 `p0` 可以正常进入临界区，且满足互斥条件。

我们需要考虑的是两个进程同时试图访问临界资源的情况。注意 `turn` 变量的含义：进程在试图访问时，首先设置自己的 `flag` 变量为 `true`，表示希望访问；但又设置 `turn` 变量为对方的进程编号，表示“谦让”，因为在循环判断条件中 `turn` 变量不是自己编号时就循环等待。这时两个进程就会互相“谦让”一番，但是这不会造成饥饿的局面，因为 `turn` 变量会有一个最终值，所以必定有进程可以结束循环进入临界区。实际的情况是，先作出“谦让”的进程先进入临界区，后作出“谦让”的进程则需要循环等待。

其实这里可以想象为两个人进门，每个人进门前都会和对方客套一句“你走先”。如果进门时没别人，就当和空气说句废话，然后大步登门入室；如果两人同时进门，就互相请先，但各自只客套一次，所以先客套的人请完对方，就等着对方请自己，然后光明正大进门。

## 28. 考查动态分区分配。

考生需对动态分区分配的四种算法加以理解。最佳适配算法是指：每次为作业分配内存空间时，总是找到能满足空间大小需要的最小的空闲分区给作业。可以产生最小的内存空闲分区。下图显示了这个过程的主存空间的变化：

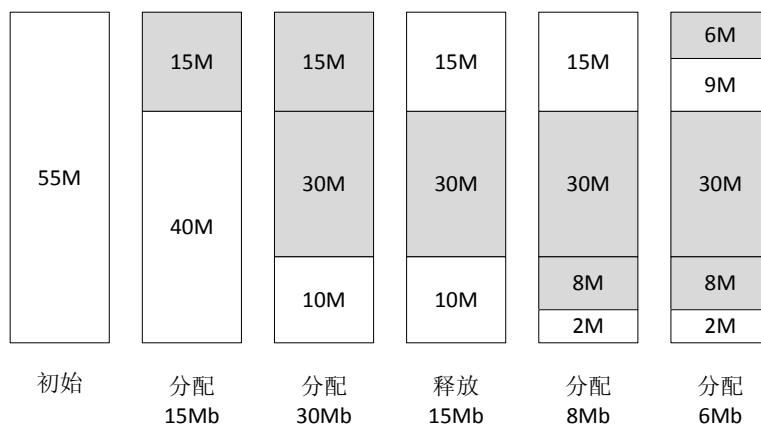


图 2010-28 最佳适配算法分配示意图

图中，灰色部分为分配出去的空间，白色部分为空闲区。这样，容易发现,此时主存中最大空闲分区的大小为 9Mb。

## 29. 考查非连续分配的分页存储管理方式。

页大小为  $2^{10}$  字节，页表项大小为 2 字节，采用二级页表，一页可存放  $2^9$  个页表项，逻辑地址空间大小为  $2^{16}$  页，要使表示整个逻辑地址空间的页目录表中包含的个数最少，则需要  $2^{16}/2^9 = 2^7 = 128$  个页面保存页表项，即页目录表中包含的个数最少为 128。

## 30. 考查磁盘文件的大小性质。

因每个磁盘索引块和磁盘数据块大小均为 256 字节。所以 4 个直接地址索引指向的数据块大小为  $4 \times 256$  字节。2 个一级间接索引共包括  $2 \times (256 \div 4)$  个直接地址索引，既其指向的数据块大小为  $2 \times (256 \div 4) \times 256$  字节。1 个二级间接地址索引所包含的直接地址索引数为  $(256 \div 4) \times (256 \div 4)$ ，即其所指向的数据块大小为  $(256 \div 4) \times (256 \div 4) \times 256$  字节。即 7 个地址项所指向的数据块总大小为  $4 \times 256 + 2 \times (256 \div 4) \times 256 + (256 \div 4) \times (256 \div 4) \times 256 = 1082368$  字节 = 1057KB。

## 31. 考查当前目录的作用。

一个文件系统含有许多级时，每访问一个文件，都要使用从树根开始直到树叶为止，包括各中间节点名的全路径名。当前目录又称工作目录，进程对各个文件的访问都相对于当前目录进行，所以检索速度要快于检索全路径名。

## 32. 考查中断处理。

键盘是典型的通过中断 I/O 方式工作的外设，当用户输入信息时，计算机响应中断并通过中断处理程序获得输入信息。

## 33. 考查计算机网络体系结构的基本概念。

我们把计算机网络的各层及其协议的集合称为体系结构。因此 A、B、D 正确，而体系结构是抽象的，它不包括各层协议及功能的具体实现细节。

## 34. 考查存储转发机制。

由题设可知，分组携带的数据长度为 980B，文件长度为 980000B，需拆分为 1000 个分组，加上头



部后，每个分组大小为 1000B，总共需要传送的数据量大小为 1MB。由于所有链路的数据传输速度相同，因此文件传输经过最短路径时所需时间最少，最短路径经过 2 个分组交换机。

当  $t = 1M \times 8 / 100Mbps = 80ms$  时，H1 发送完最后一个 bit；

由于传输延时，当 H1 发完所有数据后，还有两个分组未到达目的地，其中最后一个分组，需经过 2 个分组交换机的转发，在两次转发完成后，所有分组均到达目的主机。每次转发的时间为  $t_0 = 1K \times 8 / 100Mbps = 0.08ms$ 。

所以，在不考虑分组拆装时间和等待延时的情况下，当  $t = 80ms + 2t_0 = 80.16ms$  时，H2 接收完文件，即所需的时间至少为 80.16ms。

### 35. 考查 RIP 路由协议。

RIP 使用距离向量算法的工作过程参见内容精讲部分。

R1 在收到信息并更新路由表后，若需要经过 R2 到达 net1，则其跳数为 17，由于距离为 16 表示不可达，因此 R1 不能经过 R2 到达 net1，R2 也不可能到达 net1。B、C 错误，D 正确。而题目中并未给出 R1 向 R2 发送的信息，因此 A 也不正确。

### 36. 考查 ICMP 协议。

ICMP 差错报告报文有 5 种，终点不可达、源点抑制、时间超过、参数问题、改变路由（重定向），其中源点抑制是当路由器或主机由于拥塞而丢弃数据报时，就向源点发送源点抑制报文，使源点知道应当把数据报的发送速率放慢。

### 37. 考查子网划分与子网掩码、CIDR。

由于该网络的 IP 地址为 192.168.5.0/24，因此其网络号为前 24 位。第 25-32 位为子网位+主机位。而子网掩码为 255.255.255.248，其第 25-32 位的 248 用二进制表示为 11111000，因此后 8 位中，前 5 位用于子网号，后 3 位用于主机号。

RFC 950 文档规定，对分类的 IPv4 地址进行子网划分时，子网号不能为全 1 或全 0。但随着无分类域间路由选择 CIDR 的广泛使用，现在全 1 和全 0 的子网号也可以使用，但一定要谨慎使用，要弄清你的路由器所有的路由选择软件是否支持全 0 或全 1 的子网号这种用法。但不论是分类的 IPv4 地址还是无分类域间路由选择 CIDR，其子网中的主机号均不能为全 1 或全 0。因此该网络空间的最大子网个数为  $2^5 = 32$  个，每个子网内的最大可分配地址个数为  $2^3 - 2 = 6$  个。

### 38. 考查网络设备与网络风暴。

物理层设备中继器和集线器既不隔离冲突域也不隔离广播域；网桥可隔离冲突域，但不隔离广播域；网络层的路由器既隔离冲突域，也隔离广播域；VLAN 即虚拟局域网也可隔离广播域。对于不隔离广播域的设备，他们互连的不同网络都属于同一个广播域，因此扩大了广播域的范围，更容易产生网络风暴。

### 39. 考查 TCP 流量控制与拥塞控制。

发送方的发送窗口的上限值应该取接收方窗口和拥塞窗口这两个值中较小的一个，于是此时发送方的发送窗口为  $\min\{4000, 2000\} = 2000$  字节，由于发送方还没有收到第二个最大段的确认，所以此时主机甲还可以向主机乙发送的最大字节数为  $2000 - 1000 = 1000$  字节。

### 40. 考查 DNS 系统域名解析过程。

当采用递归查询的方法解析域名时，如果主机所询问的本地域名服务器不知道被查询域名的 IP 地址，那么本地域名服务器就以 DNS 客户的身份，向其他根域名服务器继续发出查询请求报文，这种方法用户主机和本地域名服务器发送的域名请求条数均为 1 条。