

## 参考答案

### 一、单项选择题（后附有解析）

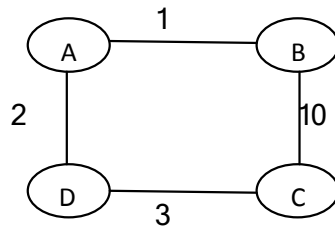
- |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. B  | 2. C  | 3. D  | 4. B  | 5. C  | 6. B  | 7. A  | 8. D  |
| 9. A  | 10. B | 11. C | 12. D | 13. D | 14. C | 15. D | 16. C |
| 17. A | 18. A | 19. D | 20. B | 21.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

### 二、综合应用题

#### 41. 解答：

该方法不一定能（或不能）求得最短路径。

例如，对于下图所示的带权图，如果按照题中的原则，从 A 到 C 的最短路径是 A->B->C，事实上其最短路径是 A->D->C。



#### 42. 解答：

##### （1）算法的基本设计思想（5分）：

问题的关键是设计一个尽可能高效的算法，通过链表的一趟遍历，找到倒数第 k 个结点的位置。算法的基本设计思想是：定义两个指针变量 p 和 q，初始时均指向头结点的下一个结点（链表的第一个结点）。p 指针沿链表移动；当 p 指针移动到第 k 个结点时，q 指针开始与 p 指针同步移动；当 p 指针移动到最后一个结点时，q 指针所指示结点为倒数第 k 个结点。以上过程对链表仅进行一遍扫描。

##### （2）算法的详细实现步骤（5分）：

① count=0，p 和 q 指向链表表头结点的下一个结点；

② 若 p 为空，转⑤；

③ 若 count 等于 k，则 q 指向下一个结点；否则，count=count+1；

④ p 指向下一个结点，转②；

⑤ 若 count 等于 k，则查找成功，输出该结点的 data 域的值，返回 1；否则，说明 k 值超过了线性表的长度，查找失败，返回 0；

⑥ 算法结束。

##### （3）算法实现（5分）：

```
typedef int ElemType;           //链表数据的类型定义
typedef struct LNode{           //链表结点的结构定义
    ElemType    data;           //结点数据
    struct LNode *link;         //结点链接指针
} *LinkList;

int Search_k(LinkList list, int k){
    //查找链表list倒数第k个结点，并输出该结点data域的值
    LinkList p = list->link, q = list->link; //指针p、q指示第一个结点
```

```

int count = 0;
while(p != NULL){      //遍历链表直到最后一个结点
    if(count < k) count++;      //计数, 若 count < k 只移动 p
    else q = q->link; p = p->link;      //之后让 p、q 同步移动
} //while
if(count < k)
    return 0;      //查找失败返回 0
else { //否则打印并返回 1
    printf("%d", q->data);
    return 1;
}
} //Search_k

```

**提示：**算法程序题，如果能够写出数据结构类型定义、正确的算法思想都会至少给一半以上分数，如果能用伪代码写出自然更好，比较复杂的地方可以直接用文字表达。

若所给出的算法采用一遍扫描方式就能得到正确结果，可给满分 15 分；若采用两遍或多遍扫描才能得到正确结果的，最高分 10 分。若采用递归算法得到正确结果的，最高给 10 分；若实现算法的空间复杂度过高（使用了大小与 k 有关的辅助数组），但结果正确，最高给 10 分。

#### 43. 解答：

(1) 按题意，外设每秒传送 0.5MB，中断时每次传送 4B。中断方式下，CPU 每次用于数据传送的时钟周期为： $5 \times 18 + 5 \times 2 = 100$ 。

为达到外设 0.5MB/s 的数据传输率，外设每秒申请的中断次数为： $0.5\text{MB}/4\text{B} = 125\,000$ 。

1 秒钟内用于中断的开销： $100 \times 125\,000 = 12\,500\,000 = 12.5\text{M}$  个时钟周期。

CPU 用于外设 I/O 的时间占整个 CPU 时间的百分比： $12.5\text{M}/500\text{M} = 2.5\%$ 。

(2) 当外设数据传输率提高到 5MB/s 时改用 DMA 方式传送，每次 DMA 传送 5 000B，1 秒钟内需产生的 DMA 次数： $5\text{MB}/5\,000\text{B} = 1\,000$ 。

CPU 用于 DMA 处理的总开销： $1\,000 \times 500 = 500\,000 = 0.5\text{M}$  个时钟周期。

CPU 用于外设 I/O 的时间占整个 CPU 时间的百分比： $0.5\text{M}/500\text{M} = 0.1\%$ 。

#### 44. 解答：

一条指令的执行过程通常由取指、译码和执行 3 个步骤完成，本题中取指用 3 个节拍、译码用 1 个节拍，执行加法运算并把结果写入主存如何完成呢？包括划分执行步骤、确定完成的功能、要提供的控制信号，这是本题要测试的内容。为回答这个问题，首先要看清图中给出的部件组成情况和信息传送的路径。

要完成的功能时  $(R0) + ((R1)) \rightarrow (R1)$ ，从图中看到：

(1) R0、R1 都有送自己内容到内总线的路径，控制信号分别是 R0out 和 R1out；

(2) ALU 加运算，2 个数据由工作寄存器 A 和内总线提供，控制信号是 Add；A 只接收内总线的内容，控制信号是 Ain；结果需存 AC，控制信号是 ACin；AC 的内容可送内总线，控制信号是 ACout；

(3) PC 可接收内总线的内容，还可增 1，控制信号是 PCin 和 PC+1，PC 的内容可送内总线，控制信号是 PCout；

(4) 指令寄存器 IR 可接收内总线的内容，控制信号是 IRin；

(5) 读写存储器时，地址由 MAR 经 AB 提供，MAR 只接收总线上的信息，控制信号是 MARin；

(6) 读存储器，提供读命令 MemR，并通过 DB 送入 MDR，控制信号是 MDRinE；MDR 的内容可送入总线，控制信号是 MDRout；

(7) 写存储器，提供写命令 MemW，数据由 MDR 通过 DB 送到存储器的数据引脚，控制信号是 MDRoutE；

然后是划分执行步骤、确定每一步完成的功能、需要提供的控制信号。这是由指令应完成的功能和计算机硬件的实际组成情况和信息传送的可用路径共同决定的，基本原则是步骤越少越好。硬件电路要能支持，可以有多种方案，解题时应参照以给出的答题格式，即取指和译码阶段的那张表的内容，但不必把表已有的内容再抄一遍。

划分指令执行步骤，确定每一步完成的功能、给出需要提供的控制信号：

请注意， $(R0) + ((R1))$  表示： $R0$  寄存器的内容与  $R1$  作地址从主存中读出来的数据完成加法运算；而  $\rightarrow (R1)$  表示把  $R1$  的内容作为主存储器的地址完成写主存操作。为防止出现误解，题中还特地对此作了文字说明。这条指令的功能是先到主存储器取一个数，之后运算，再将结果写回主存储器。

(1) 执行相加运算，需把存储器中的数据读出，为此首先送地址，将  $R1$  的内容送  $MAR$ ，控制信号是  $R1out$ 、 $MARin$ 。

(2) 启动读主存操作，读出的内容送入  $MDR$ ，控制信号是  $MemR$ 、 $MDRinE$ 。还可同时把  $R0$  的内容经内总线送入  $A$ ，用到的控制信号是  $R0out$ 、 $Ain$ 。

(3) 执行加法运算，即  $A$  的内容与  $MDR$  的内容相加，结果保存到  $AC$ ，控制信号是  $MDRout$ 、 $Add$ 、 $ACin$ 。

(4) 要把  $AC$  的内容写入主存，由于  $R1$  的内容已经在  $MAR$  中，地址已经有了，但需要把写入的数据（已经在  $AC$  中）经内总线送入  $MDR$ ，控制信号是  $ACout$ 、 $MDRin$ 。

(5) 给出写主存的命令，把  $MDR$  的内容经  $DB$  送存储器的数据线引脚，执行写操作，控制信号是  $MDRoutE$ 、 $MemW$ 。

这几个步骤是有先后次序的，前面的完成了，下一步才可以执行，也保证了不会产生硬件线路的冲突。请注意，使用最为频繁的是内总线，它在任何时刻只能接收一个输入数据，并且向内总线发送信息的电路只能以三态门器件连接到内总线，5 个向内总线发送信息的控制信号（ $ACout$ ， $PCout$ ， $R0out$ ， $R1out$ ， $MDRout$ ）最多只能有一个为 1，其他 4 个必须全为 0，或者 5 个全为 0。

仔细看一下，发现可以把第 2 个步骤的操作划分到两个步骤中完成，一个步骤中安排  $MDR$  接收从存储器中读出的内容，到另外一个步骤实现  $R0$  的内容送入  $A$ ，这多用了个操作步骤，指令的执行速度会变慢。有些解题者在写存储器之前，还会再执行一次把  $R1$  的内容送  $MAR$ ，尽管无此必要，但不属于原理上的错误。

**当然还可以有其他的设计结果。**

解题时这些叙述内容不必写出来（这里写出这些内容是帮助大家领会本题要测试的知识点和指令的执行过程），直接按照已经给出的表格的形式、按照提供的填写办法把设计的表格及其内容填好就可以了。

请注意，题目表格内容（告诉你答题的格式和答题内容的表达方式）与你答题的表格内容合在一起才是这条指令的完整的执行过程，千万不要产生任何错觉。

**参考答案一：**

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	$R1out, MARin$
C6	$MDR \leftarrow M(MAR)$ $A \leftarrow (R0)$	$MemR, MDRinE,$ $R0out, Ain$
C7	$AC \leftarrow (MDR) + (A)$	$MDRout, Add, ACin$
C8	$MDR \leftarrow (AC)$	$ACout, MDRin$
C9	$M(MAR) \leftarrow (MDR)$	$MDRoutE, MemW$

“ $A \leftarrow (R0)$ ”也可在 C7：“ $AC \leftarrow (MDR) + (A)$ ”之前单列的一个时钟周期内执行。

**参考答案二：**

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	$R1out, MARin$

C6	$MDR \leftarrow M(MAR)$	MemR,MDRinE
C7	$A \leftarrow (MDR)$	MDRout,Ain
C8	$AC \leftarrow (A)+(R0)$	R0out,Add,ACin
C9	$MDR \leftarrow (AC)$	ACout,MDRin
C10	$M(MAR) \leftarrow (MDR)$	MDRoutE,MemW

45. 解答:

定义信号量 odd 控制 P1 与 P2 之间的同步; even 控制 P1 与 P3 之间的同步; empty 控制生产者与消费者之间的同步; mutex 控制进程间互斥使用缓冲区。程序如下:

```
semaphore odd = 0, even = 0, empty = N, mutex = 1;

P1( )
{
    x = produce();    //生成一个数
    P(empty);          //判断缓冲区是否有空单元
    P(mutex);          //缓冲区是否被占用
    Put();
    V(mutex);          //释放缓冲区
    if(x%2 == 0)
        V(even);        //如果是偶数, 向 P3 发出信号
    else
        V(odd);         //如果是奇数, 向 P2 发出信号
}

P2( )
{
    P(odd);             //收到 P1 发来的信号, 已产生一个奇数
    P(mutex);           //缓冲区是否被占用
    getodd();
    V(mutex);           //释放缓冲区
    V(empty);           //向 P1 发信号, 多出一个空单元
    countodd();
}

P3( )
{
    P(even);            //收到 P1 发来的信号, 已产生一个偶数
    P(mutex);           //缓冲区是否被占用
    geteven();
    V(mutex);           //释放缓冲区
    V(empty);           //向 P1 发信号, 多出一个空单元
    counteven();
}
```

46. 解答:

(1) 根据页式管理的工作原理, 应先考虑页面大小, 以便将页号和页内位移分解出来。页面大小为 4KB,

即  $2^{12}$ ，则得到页内位移占虚地址的低 12 位，页号占剩余高位。可得三个虚地址的页号 P 如下（十六进制的一位数字转换成 4 位二进制，因此，十六进制的低三位正好为页内位移，最高位为页号）：

2362H: P=2，访问快表 10ns，因初始为空，访问页表 100ns 得到页框号，合成物理地址后访问主存 100ns，共计 10ns+100ns+100ns=210ns。

1565H: P=1，访问快表 10ns，落空，访问页表 100ns 落空，进行缺页中断处理  $10^8$ ns，访问快表 10ns，合成物理地址后访问主存 100ns，共计 10ns+100ns+ $10^8$ ns+10ns+100ns=100 000 220ns。

25A5H: P=2，访问快表，因第一次访问已将该页号放入快表，因此花费 10ns 便可合成物理地址，访问主存 100ns，共计 10ns+100ns=110ns。

(2) 当访问虚地址 1565H 时，产生缺页中断，合法驻留集为 2，必须从页表中淘汰一个页面，根据题目的置换算法，应淘汰 0 号页面，因此 1565H 的对应页框号为 101H。由此可得 1565H 的物理地址为 101565H。

**47. 解答：**

(1) CIDR 中的子网号可以全 0 或全 1，但主机号不能全 0 或全 1。

因此若将 IP 地址空间 202.118.1.0/24 划分为 2 个子网，且每个局域网需分配的 IP 地址个数不少于 120 个，子网号至少要占用一位。

由  $2^6-2 < 120 < 2^7-2$  可知，主机号至少要占用 7 位。

由于源 IP 地址空间的网络前缀为 24 位，因此 主机号位数+子网号位数=8。

综上可得主机号位数为 7，子网号位数为 1。

因此子网的划分结果为：子网 1：202.118.1.0/25，子网 2：202.118.1.128/25。

地址分配方案：子网 1 分配给局域网 1，子网 2 分配给局域网 2，或子网 1 分配给局域网 2，子网 2 分配给局域网 1。

(2) 由于局域网 1 和局域网 2 分别与路由器 R1 的 E1、E2 接口直接相连，因此在 R1 的路由表中，目的网络为局域网 1 的转发路径是直接通过接口 E1 转发，目的网络为局域网 2 的转发路径是直接通过接口 E1 转发。由于局域网 1、2 的网络前缀均为 25 位，因此它们的子网掩码均为 255.255.255.128。

根据题意，R1 专门为域名服务器设定了一个特定的路由表项，因此该路由表项中的子网掩码应为 255.255.255.255。对应的下一跳转发地址是 202.118.2.2，转发接口是 L0。

根据题意，到互联网的路由实质上相当于一个默认路由，默认路由一般写作 0/0，即目的地址为 0.0.0.0，子网掩码为 0.0.0.0。对应的下一跳转发地址是 202.118.2.2，转发接口是 L0。

综上可得到路由器 R1 的路由表为：

（若子网 1 分配给局域网 1，子网 2 分配给局域网 2）

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
202.118.1.0	255.255.255.128	—	E1
202.118.1.128	255.255.255.128	—	E2
202.118.3.2	255.255.255.255	202.118.2.2	L0
0.0.0.0	0.0.0.0	202.118.2.2	L0

（若子网 1 分配给局域网 2，子网 2 分配给局域网 1）

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
202.118.1.128	255.255.255.128	—	E1
202.118.1.0	255.255.255.128	—	E2
202.118.3.2	255.255.255.255	202.118.2.2	L0
0.0.0.0	0.0.0.0	202.118.2.2	L0

---

(3) 局域网 1 和局域网 2 的地址可以聚合为 202.118.1.0/24，而对于路由器 R2 来说，通往局域网 1 和 2 的转发路径都是从 L0 接口转发，因此采用路由聚合技术后，路由器 R2 到局域网 1 和局域网 2 的路由为：

目的网络 IP 地址	子网掩码	下一跳 IP 地址	接口
202.118.1.0	255.255.255.0	202.118.2.1	L0

**2009 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——选择题部分解析**

**一、单项选择题**

**1. 考查栈和队列的特点及应用。**

C 和 D 直接排除, 缓冲区的特点需要先进先出, 若用栈, 则先进入缓冲区的数据则要排队到最后才能打印, 不符题意, 所以只有队列符合题意。

**2. 考查栈的最大递归深度。**

时刻注意栈的特点是先进后出。下面是出入栈的详细过程:

序号	说明	栈内	栈外	序号	说明	栈内	栈外
1	a 入栈	a		8	e 入栈	ae	bdc
2	b 入栈	ab		9	f 入栈	aef	bdc
3	b 出栈	a	b	10	f 出栈	ae	bdcf
4	c 入栈	ac	b	11	e 出栈	a	bdcfe
5	d 入栈	acd	b	12	a 出栈		bdcfea
6	d 出栈	ac	bd	13	g 入栈	g	bdcfea
7	c 出栈	a	bdc	14	g 出栈		bdcfeag

栈内的最大深度为 3, 故栈 S 的容量至少是 3。

**3. 考查二叉树的特殊遍历。**

分析遍历后的结点序列, 可以看出根结点是在中间被访问的, 而且右子树结点在左子树之前, 则遍历的方法是 RNL。本题考查的遍历方法并不是二叉树遍历的三种基本遍历方法, 对于考生而言, 重要的是要掌握遍历的思想。

**4. 考查平衡二叉树的定义。**

根据平衡二叉树的定义有, 任意结点的左右子树高度差的绝对值不超过 1。而其余三个答案均可以找到不符合的结点。

**5. 考查完全二叉树的特点。**

完全二叉树比起满二叉树只是在最下面一层的右边缺少了部分叶结点, 而最后一层之上是个满二叉树, 并且只有最后两层上有叶结点。第 6 层有叶结点则完全二叉树的高度可能为 6 或 7, 显然树高为 7 时结点更多。若第 6 层上有 8 个叶结点, 则前六层为满二叉树, 而第 7 层缺失了  $8 \times 2 = 16$  个叶结点, 故完全二叉树的结点个数最多为  $2^7 - 1 - 16 = 111$  个结点。

**6. 考查森林和二叉树的转换。**

森林与二叉树的转换规则为“左孩子右兄弟”。在最后生成的二叉树中, 父子关系在对应森林关系中可能是兄弟关系或原本就是父子关系。

情形 I: 若结点 v 是结点 u 的第二个孩子结点, 在转换时, 结点 v 就变成结点 u 第一个孩子的右孩子, 符合要求。

情形 II: 结点 u 和 v 是兄弟结点的关系, 但二者之中还有一个兄弟结点 k, 则转换后, 结点 v 就变为结点 k 的右孩子, 而结点 k 则是结点 u 的右孩子, 符合要求。

情形 III: 结点 v 的父结点要么是原先的父结点或兄弟结点。若结点 u 的父结点与 v 的父结点是兄弟关系, 则转换之后, 不可能出现结点 u 是结点 v 的父结点的父结点。

**7. 考查无向连通图的特性。**

每条边都连接了两个结点, 则在计算顶点的度之时, 这条边都被计算了两次, 故所有顶点的度之和

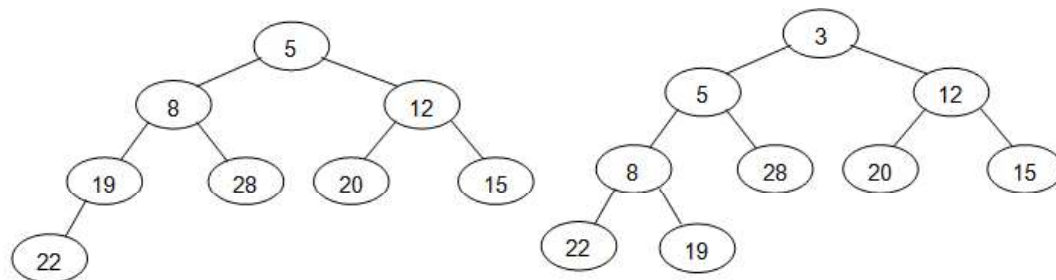
为边数的两倍，显然必为偶数。而 ii 和 iii 则不一定正确，如：对顶点数  $N \geq 1$  无向完全图不存在一个顶点的度为 1，并且边数与顶点数的差要大于 1。

8. 考查 m 阶 B-树的定义。

A、B 和 C 都是 B-树的特点，而选项 D 则是 B+树的特点。注意区别 B-树和 B+树各自的特点。

9. 考查小根堆的调整操作。

小顶堆在逻辑上可以用完全二叉树来表示，根据关键序列得到的小顶堆的二叉树形式为下图左图：



插入关键字 3 时，先将其放在小顶堆的末端，再将该关键字向上进行调整，得到的结果上图右边所示。所以，调整后的小顶堆序列为：3，5，12，8，28，20，15，22，19。

10. 考查各排序算法的特点。

解答题之前要对不同排序算法的特点极为清楚。对于起泡排序和选择排序而言，每一趟过后都能确定一个元素的最终位置，而由题目中所说，前两个元素和后两个元素均不是最小或最大的两个元素并按序排列。答案 D 中的二路归并排序，第一趟排序结束都可以得到若干个有序子序列，而此时的序列中并没有两两元素有序排列。插入排序在每趟排序结束后能保证前面的若干元素是有序的，而此时第二趟排序后，序列的前三个元素是有序的，符合其特点。

11. 考查指令执行过程。

通常完成一条指令可分为取指阶段和执行阶段。在取指阶段通过访问存储器可将指令取出；在执行阶段通过访问存储器可以将操作数取出。这样，虽然指令和数据都是以二进制代码形式存放在存储器中，但 CPU 可以判断在取指阶段访问存储器取出的二进制代码是指令；在执行阶段访存取出的二进制代码是数据。

12. 考查符号位的扩展。

结合题干及选项可知，int 为 32 位，short 为 16 位；又 C 语言的数据在内存中为补码形式，故 x、y 的机器数写为 0000007FH、FFF7H；

执行  $z=x+y$  时，由于 x 是 int 型，y 为 short 型，故需将 y 的类型强制转换为 int，在机器中通过符号位扩展实现，由于 y 的符号位为 1，故在 y 的前面添加 16 个 1，即可将 y 强制转换为 int 型，其十六进制形式为 FFFFFFF7H；

然后执行加法，即  $0000007FH + FFFFFFF7H = 00000076H$ ，其中最高位的进位 1 自然丢弃。故选 D。

13. 考查浮点加法运算。

根据题意，X 可记为 00, 111; 00, 11101 (分号前为阶码，分号后为尾数)，Y 可记为 00, 101; 00, 10100。

首先对阶，X、Y 阶码相减，即  $00, 111 - 00, 101 = 00, 111 + 11, 0111 = 00, 010$ ，可知 X 的阶码比 Y 的阶码大 2，根据小阶向大阶看齐的原则，将 Y 的阶码加 2，尾数右移 2 位，可得 Y 为 00, 111; 00, 00101。

尾数相加，即  $00, 11101 + 00, 00101 = 01, 00010$ ，尾数相加结果符号位为 01，故需进行右规。

规格化，将尾数右移 1 位，阶码加 1，得 X+Y 为 01, 000; 00, 1000，阶码符号位为 01，说明发生溢出。

14. 考查 Cache 与主存之间的映射方式。



由于 Cache 共有16块,采用2路组相联,因此共有8组,0,1,2,...,7。并且主存的某一字块按模8映像到 Cache 某组的任一字块中,即主存的第0,8,16...字块可以映像到 Cache 第0组2个字块的任一字块中,而129号单元是位于第4块主存块中,因此将映射到 Cache 第4组2个字块的任一字块中。

**注意:** 由于在计算机系统结构中和计算机组成原理的某些教材中介绍的组相联跟此处的组相联并不相同,导致部分考生理解错题目。考生应以真题为准,以后再出现类似题目,应以此种解答为标准。

#### 15. 考查存储器的扩展。

首先确定 ROM 的个数,ROM 区为4KB,选用2K×8的 ROM 芯片,需要  $\frac{4K \times 8}{2K \times 8} = 2$  片,采用字扩展方式;60KB 的 RAM 区,选用4K×4的 RAM 芯片,需要  $\frac{60K \times 8}{4K \times 4} = 30$  片,采用字和位同时扩展方式。

#### 16. 考查相对寻址。

相对寻址 EA=(PC)+A, 首先要求的是取指令后 PC 的值。转移指令由两个字节组成,每取一个字节 PC 自动加 1,因此取指令后 PC 值为 2002H,故 EA=(PC)+A=2002H+06H=2008H。

#### 17. 考查 RISC 的特性。

相对于 CISC 计算机,RISC 计算机的特点指令条数少;指令长度固定,指令格式和寻址种类少;只有取数/存数指令访问存储器,其余指令的操作均在寄存器之间进行;CPU 中通用寄存器多;大部分指令在一个或者小于一个机器周期内完成;以硬布线逻辑为主,不用或者少用微程序控制。

#### 18. 考查流水线中时钟周期的特性。

时钟周期应以最长的执行时间为准,否则用时长的流水段的功能将不能正确完成。

#### 19. 考查硬布线控制器的特点。

硬布线控制器的速度取决于电路延迟,所以速度快;微程序控制器采用了存储程序原理,每条指令都要访控存,所以速度慢。硬布线控制器采用专门的逻辑电路实现,修改和扩展困难。

#### 20. 考查总线的基本概念。

总线带宽是指单位时间内总线上可传输数据的位数,通常用每秒钟传送信息的字节数来衡量,单位可用字节/秒(Bps)表示。根据题意可知,在  $2 \times 10^{10}$ Hz 秒内传输了 4B,所以  $4B \times 10^{10} \text{Hz} / 2 = 20 \text{MB/S}$ 。

#### 21. 考查 Cache 的命中率。

命中率=Cache 命中的次数/所有访问次数,有了这个公式这道题就很容易看出,要注意的一点是看清题,题中说明的是缺失50次,而不是命中50次,仔细审题是做对题的第一步。

#### 22. 考查中断的分类。

选项中能引起外部中断的只能是输入设备键盘。

#### 23. 考查并行性的限定。

单处理机系统中只有一条指令流水线,一个多功能的操作部件,每个时钟周期只能完成一条指令,故进程与进程显然不可以并行。

#### 24. 考查几种基本的调度算法概念。

高响应比优先调度算法,同时考虑每个进程的等待时间和需要的执行时间,从中选出响应比最高的进程投入执行。响应比R定义如下:

响应比  $R = (\text{等待时间} + \text{执行时间}) / \text{执行时间}$

#### 25. 考查死锁的条件。

这种题用到组合数学中鸽巢原理的思想,考虑最极端情况,因为每个进程最多需要 3 台打印机,如果每个进程已经占有了两个打印机,那么只要还有多的打印机,那么总能满足达到 3 台的条件,所以,将 8 台打印机分给 K 个进程,每个进程有 2 台打印机,这个情况就是极端情况,K 为 4。

**26. 考查分区分配存储管理方式的保护措施。**

分区分配存储管理方式的保护措施是设置界地址寄存器。每个进程都有自己独立的进程空间，如果一个进程在运行时所产生的地址在其地址空间之外，则发生地址越界，即当程序要访问某个内存单元时，由硬件检查是否允许，如果允许则执行，否则产生地址越界中断，由操作系统进行相应处理。

**27. 考查分段存储管理系统。**

段地址为 32 位二进制数，其中 8 位表示段号，则段内位移占用  $32-8=24$  位二进制数，故最大段长为  $2^{24}$  字节。

**28. 考查文件物理结构的特性。**

随机访问是索引结构的特性。

**29. 考查磁盘的调度算法。**

类似于电梯调度的思想。首先，磁头选择与当前磁头所在磁道距离最近的请求作为首次服务的对象（110），当磁头沿途相应访问请求序列直到达到一端末（110,170,180,195），再反向移动响应另一端的访问请求（68,45,35,12）。

**30. 考查文件控制块的内容。**

在文件控制块中，通常含有以下 3 类信息，即基本信息、存取控制信息及使用信息。

**31. 考查软/硬链接建立的属性。**

建立符号链接（软链接）时，引用计数值直接复制；建立硬链接时，引用计数值加 1。删除文件时，删除操作对于符号链接是不可见的，这并不影响文件系统，当以后再通过符号链接访问时，发现文件不存在，直接删除符号链接；但是对于硬链接则不可以直接删除，引用计数值减 1，若值不为 0，则不能删除此文件，因为还有其它硬链接指向此文件。

**32. 考查系统调用的设备标识。**

用户程序对 I/O 设备的请求采用逻辑设备名，而在程序实际执行时使用物理设备名。

**33. 考查 OSI 模型中传输层的功能。**

传输层提供应用进程间的逻辑通信，即端到端的通信。而网络层提供点到点的逻辑通信。因此选 B。

**34. 考查奈氏准则和香农定理。**

采用 4 个相位，每个相位有 4 种幅度的 QAM 调制方法，每个信号可以有 16 种变化，传输 4bit 的数据。根据奈奎斯特定理，信息的最大传输速率为  $2 \times 3K \times 4 = 24Kbps$ 。

**35. 考查后退 N 帧协议的工作原理。**

在后退 N 帧协议中，发送方可以连续发送若干个数据帧，如果收到接收方的确认帧则可以继续发送。若某个帧出错，接收方只是简单的丢弃该帧及其后所有的后续帧，发送方超时后需重传该数据帧及其后续的所有数据帧。这里要注意，连续 ARQ 协议中，接收方一般采用累积确认的方式，即接收方对按序到达的最后一个分组发送确认，因此题目中收到 3 的确认帧就代表编号为 0、1、2、3 的帧已接收，而此时发送方未收到 1 号帧的确认只能代表确认帧在返回的过程中丢失了，而不代表 1 号帧未到达接收方。因此需要重传的帧为编号是 4、5、6、7 的帧。

**36. 考查交换机的工作原理。**

交换机实质上是一个多端口网桥，工作在数据链路层，数据链路层使用物理地址进行转发，而转发通常都是根据目的地址来决定出端口。

**37. 考查 CSMA/CD 协议的工作原理。**

首先由例 8 可知，若最短帧长减少，而数据传输速率不变，则需要使冲突域的最大距离变短来实现争用期的减少。争用期是指网络中收发结点间的往返时延，因此假设需要减少的最小距离为 s，单位是 m，则可以得到下式（注意单位的转换）： $2 \times [s / (2 \times 10^8)] = 800 / (1 \times 10^9)$ ，因此可得  $s = 80$ ，即最远的两个站点

之间的距离最少需要减少 80m。

**38. 考查 TCP 的数据编号与确认。**

TCP 是面向字节流的，其选择确认（Selective ACK）机制是接收端对字节序号进行确认，其返回的序号是接收端下一次期望接收的序号，因此主机乙接收两个段后返回给主机甲的确认序列号是 1000。

**39. 考查 TCP 的拥塞控制方法。**

本题计算原理如图 4 所示。无论在慢开始阶段还是在拥塞避免阶段，只要发送方判断网络出现拥塞（其根据就是没有按时收到确认），就要把慢开始门限 **ssthresh** 设置为出现拥塞时的发送方窗口值的一半（但不能小于 2）。然后把拥塞窗口 **cwnd** 重新设置为 1，执行慢开始算法。这样做的目的就是要迅速减少主机发送到网络中的分组数，使得发生拥塞的路由器有足够时间把队列中积压的分组处理完毕。

因此，在发送拥塞后，慢开始门限 **ssthresh** 变为  $16/2 = 8 \text{ KB}$ ，发送窗口变为 1 KB。在接下来的 3 个 RTT 内，拥塞窗口执行慢开始算法，呈指数形式增加到 8 KB，此时由于慢开始门限 **ssthresh** 为 8 KB，因此转而执行拥塞避免算法，即拥塞窗口开始“加法增大”。因此第 4 个 RTT 结束后，拥塞窗口的大小为 9 KB。

**40. 考查 FTP 协议的特点。**

FTP 协议是基于传输层 TCP 协议的。FTP 的控制连接使用端口 21，用来传输控制信息（如连接请求，传送请求等），数据连接使用端口 20，用来传输数据。