



## 2023 年全国硕士研究生招生考试 408 计算机超级密押卷三

## 参考答案与解析

一、单项选择题：1~40 小题，每小题 2 分，共 80 分，下列每小题给出的四个选项中，只有一个选项是符合题目要求的。请在答题卡上将所选项的字母涂黑。

1. 【参考答案】A。

【解析】本题考查时间复杂度。将算法中基本运算的执行次数的数量级作为时间复杂度。基本运算是“ $i=i/2;$ ”，设其执行次数为  $k$ ，则  $(n \times n)/(2^k)=1$ ，得  $k=\log_2 n^2$ ，因此  $k=\log_2 n^2=2 \log_2 n$ ，即  $k$  的数量级为  $\log_2 n$ ，因此时间复杂度为  $O(\log_2 n)$ 。

2. 【参考答案】C。

【解析】本题考查入栈与出栈的顺序关系。有两种方法，第一种方法是把所有出栈序列选出来并找到第二个元素为  $c$  的。第二种是模拟保证第二个出栈元素为  $c$ ，有以下几种情况：①  $a$  入栈、出栈， $b$  入栈， $c$  入栈、出栈，即出栈顺序为  $acxx$ ，此时栈中为  $b$ ，还有  $d$  未入栈， $xx$  可能为  $bd$ 、 $db$  共 2 种；②  $a$  入栈， $b$  入栈、出栈， $c$  入栈、出栈，即出栈顺序为  $bcxx$ ，此时栈中为  $a$ ，还有  $d$  未入栈， $xx$  可能为  $ad$ 、 $da$  共 2 种；③  $a$  入栈， $b$  入栈， $c$  入栈， $d$  入栈、出栈， $c$  出栈，即出栈顺序为  $dcxx$ ，此时栈中为  $ab$ ， $xx$  只能为  $ba$ ，共 1 种。总共 5 种，故选 C。

3. 【参考答案】C。

【解析】本题考查串的 next 数组。

1) 设  $next[1]=0$ ， $next[2]=1$ 。

编号	1	2	3	4	5
S	a	c	a	b	a
next	0	1			

2) 当  $j=3$  时，此时  $k=next[j-1]=next[2]=1$ ，观察  $S[2]$  与  $S[k](S[1])$  是否相等， $S[2]=c$ ， $S[1]=a$ ， $S[2] \neq S[1]$ ，此时  $k=next[k]=0$ ，所以  $next[j]=1$ 。

↓  $j-1=2$

a   c   A   b   a   a  
a   C   a   b   a  
↑  $k=1$

3) 当  $j=4$  时，此时  $k=next[j-1]=next[3]=1$ ，观察  $S[3]$  与  $S[k](S[1])$  是否相等， $S[3]=a$ ， $S[1]=a$ ， $S[3]=S[1]$ ，所以  $next[j]=k+1=2$ 。

↓  $j-1=3$

a   c   a   b   a  
a   c   a   b   a  
↑  $k=1$

4) 当  $j=5$  时，此时  $k=next[j-1]=next[4]=2$ ，观察  $S[4]$  与  $S[k](S[2])$  是否相等， $S[4]=b$ ， $S[2]=c$ ，



扫描二维码,进  
研友交流社区  
获取一手资讯

$S[4] \neq S[2]$ , 所以  $k = \text{next}[k] = 1$ 。

$\downarrow j-1=4$

a	c	a	b	a
		a	c	a
			b	a

$k=2$

5) 此时  $S[k] = S[1] = a$ ,  $S[4] \neq S[1]$ , 所以  $k = \text{next}[k] = \text{next}[1] = 0$ , 所以  $\text{next}[j] = 1$ 。

$\downarrow j-1=4$

a	c	a	b	a
		a	c	a
			b	a

$\uparrow k=1$

此时可知  $\text{next}$  数组为 01121, 故选 C。

4. 【参考答案】B。

【解析】本题考查并查集和图的算法。Kruskal 算法流程先将所有的边从小到大排序, 然后使用并查集, 每次选最小的边, 并且该边要保证其两端的两个顶点属于不同的集合, 选择完边之后进行一次并操作, 重复选边及并操作  $n-1$  次得到最小生成树。故选 B。

5. 【参考答案】A。

【解析】将森林转化为二叉树分为两步, 第一步将每一棵树转化为二叉树, 第二步, 因为每棵树根的右子树是空, 所以将第二棵树接成第一棵树的右子树; 将第三棵树接成第二棵树的右子树。于是转换成之后, 根节点的左子树的结点个数就是除了根之外的第一棵树的结点个数。

6. 【参考答案】D。

【解析】B-树的性质。我们在课程中重点强调过, 根结点是一个特殊分子。I: 针对根结点成立, 但是其他结点的子树个数是  $\lceil m/2 \rceil$ , I 错误; II: 对于 B-树来说, 结点中的关键字是不重复的, 因此虽然关键值只是起到索引作用, 但是不一定非要找到叶子结点, II 错误; III 正确; IV: 当插入关键字引起分裂, 只是将中间结点提到父结点中, 但是父结点不一定引起分裂, 只有当原父结点中的关键字个数是  $m-1$  时, 才会引起父结点分裂。同时父结点分类会将关键字提到父结点的父结点中, 但是不一定树高一层, 只有逐级分裂, 引起根节点分裂时, 才会树高一层。

7. 【参考答案】C。

【解析】首先根据题目画出图形, 然后根据图形求出该 AOE 网的关键路径便可以得出结论。

8. 【参考答案】B。

【解析】Prim 算法产生边的顺序为 (A,D)、(C,D)、(C,F)、(E,F)、(A,B), Kruskal 顺序为 (C,F)、(A,D)、(C,D)、(E,F)、(A,B), 顺序相同的是 (E,F)、(A,B), 所以是两条边。

9. 【参考答案】B。

【解析】拓扑排序的方法:



(1) 在有向图中选一个没有前驱的顶点且输出之;

(2) 从图中删除该顶点和所有以它为尾的弧。

重复上述两步,直至全部顶点均已输出或者当前图中不存在无前驱的顶点为止。后一种情况则说明有向图中存在环。由图可得到的拓扑序列为  $abced$ ,  $abecd$ ,  $aebcd$ 。

10. 【参考答案】D。

【解析】在直接插入排序方法中,若待排序列中的最后一个元素应插入到表中的第一个位置,则前面的有序子序列中的所有元素都不在最终位置上。这个结论大家记住,若待排序列中的最后一个元素应插入到表中的第一个位置,则前面的有序子序列中的所有元素都不在最终位置上的排序算法是插入类,归并类和基数类算法。

11. 【参考答案】D。

【解析】若待排序列中的最后一个元素应插入到表中的第一个位置,则前面的有序子序列中的所有元素都不在最终位置上的排序算法是插入类,归并类和基数类算法。对于希尔排序也可以排除了,因为无法找到合适的步长。通过每次选择第一个元素作为基准元素,可以实现快速排序,正确。

12. 【参考答案】B。

【解析】冯·诺依曼计算机的特点中,数据和指令以同等地位存储在存储器中,且都是以二进制的形式,因此 B 错误。为了方便人进行书写,一般表示时用十六进制。

13. 【参考答案】A。

【解析】把 2 化成二进制,得到  $0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010$ ,按位取反,得到  $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101$ ,这是补码形式,因此得到原码是  $1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0011$ ,得到-3。

14. 【参考答案】B。

【解析】45100000H 的二进制数为  $0100\ 0101\ 0001\ 0000\ 0000\ 0000\ 0000\ 0000$ ,第 1 位为符号位,表示正数,随后 8 位  $1000\ 1010$  为用移码表示的阶码,减去 127 得到十进制数 11,而 IEEE 754 中单精度数在阶码不为 0 时隐含 1,所以尾数为  $(1.0010)_2=1.125$ 。

15. 【参考答案】B。

【解析】对于选项 I: 首先,ROM 和 RAM 都是采用随机存取方式。由于 EPROM 属于 ROM,故采用随机存取方式。故 I 正确。

对于选项 II: SRAM 采用双稳态触发器来记忆信息,因此不需要刷新;而 DRAM 采用电容存储电荷的原理来存储信息,只能维持很短的时间,因此需要刷新,故 I 正确。

对于选项 III: Cache 需要有信息的输入和输出,而 ROM 只可读,不可输入,因此不能作为 Cache,故 III 错误。



扫描二维码,进  
研友交流社区  
获取一手资讯

16. 【参考答案】D。

【解析】首先算出地址结构,因为主存块大小 64B,因此块内地址是 6 位,数据区容量是 32KB,一共有  $32\text{KB}/64\text{B}=512$  块,采用 8 路组相连结构,因此有  $512/8=64$  组,所以组号是 6 位。于是地址结构为

主存标记 $32-6-6=20$ 位	组号 6 位	块内地址 6 位
--------------------	--------	----------

所以 Cache 大小为  $512 * (20+1+1) + 32\text{K} * 8 = 267\text{K}$ ,因此数据 cache 和 code cache 一共是  $267 * 2 = 534\text{k}$ 。

其中 1 表示有效位,1 表示脏位,20 表示主存标记。

17. 【参考答案】B。

【解析】首先给出解答步骤,当前指令地址为 3000,取完这条指令后,PC 的值增加一个指令字长度,即 3002,加上偏移量-5,所以执行完这条指令后,目标地址为 2997,然后将这个值覆盖到 PC 当中。

18. 【参考答案】B。

【解析】由于二地址指令操作码字段位数为 2,最多可以有 4 条二地址指令,而只使用了两条,前两位剩下两条,即多出 1 位留作扩展用,所以剩余空间为  $2^{1+3+3}=128$ ,又因为其中包含了 10 条零地址指令,所以可用的空间还有 118,在这个空间当中,由于一地址指令后三位为地址,故可设计出  $118/2^3$ ,结果取整。

19. 【参考答案】D。

【解析】流水线的执行时间,  $mt + (n-1)t$ ,其中  $m$  是流水线的级数(段数), $n$  是指令条数。但是这个题目中  $mt + (n-2)t + t = 203\text{ns}$

20. 【参考答案】D。

【解析】在流水线中会出现三种相关,影响流水线的畅通流动,这三种相关是结构相关(冒险),数据相关(冒险)和控制相关(冒险)。

(1) 结构相关是当多条指令进入流水线后,硬件资源满足不了指令重叠执行的要求时产生的解决方案是:暂停下一条指令的执行;增加硬件资源。

(2) 数据相关是指令在流水线中重叠执行时,当后继指令需要用到前面指令的执行结果时发生的。解决方案是:暂停后续指令的执行;数据旁路(专用数据通路技术):把前一条指令的执行结果通过专门的电路直接发往下一条指令。

(3) 控制相关是当流水线遇到分支指令和其他改变 PC 值的指令时引起的。解决方案是:暂停下一条指令的执行;指令预取和指令预判:提前判断是否发生转移。

21. 【参考答案】D。

【解析】可以只重点记忆后三个。其他的都是相反的。



	CISC	RISC
指令系统	复杂,庞大	简单,精简
指令数	一般大于 200	一般小于 100
指令格式	一般大于 4	一般小于 4
指令字长	一般大于 4	一般小于 4
寻址方式	不固定	固定 32 位
可访问指令	不加限制	只有 LOAD/STORE 指令
各种指令使用频率	相差很大	相差不大
各种指令执行时间	相差很大	绝大多数在一个机器周期完成
优化编译实现	很难	较容易
程序源代码长度	较短	较长
控制逻辑实现方式	绝大多数为微程序控制	绝大多数为硬连线控制

## 22. 【参考答案】D

【解析】DMA 方式下数据的传送是硬件完成的,不需要经过 CPU。DMA 请求和中断请求同时发生,应该先响应 DMA,因为 DMA 传送数据,如果不及时相应,可能会造成丢失或者失效。

## 23. 【参考答案】B。

【解析】只用于操作系统或其他系统软件,一般不直接提供给用户使用,A、D 错。设备驱动程序需要用到系统底层的操作,故需要能够执行特权指令。

## 24. 【参考答案】A。

【解析】启动服务、启动程序、用户登录均会创建新进程,而打开浏览器不会创建新进程。打开浏览器的操作适合于使用线程进行管理,它属于轻量级的操作。

## 25. 【参考答案】C。

【解析】在调度算法中,FCFS,短作业优先调度,高响应比优先调度算法,基于优先权调度算法均可以按照抢占式和非抢占式模式实现。时间片轮转调度算法和多级反馈队列调度算法是基于时间片的调度算法,当进程的时间片用完,一定会被剥夺 CPU,所以一定是可抢占算法。

## 26. 【参考答案】D。

【解析】根据题目条件,可以计算出 Need 矩阵是

进程	Need			Available		
	A	B	C	A	B	C
P1	3	4	7	2	3	2
P2	1	3	4			
P3	0	0	6			
P4	2	2	1			
P5	1	1	0			





扫描二维码,进  
研友交流社区  
获取一手资讯

所以可以让 P4 先运行完成,回收 P4 资源,之后再让 P5 运行完成,回收 P5 的资源,之后再让 P1 运行完成,回收 P1 的资源,之后再让 P2 运行完成,回收 P2 的资源,之后再让 P3 运行完成,回收 P3 的资源,因此存在安全序列,选 D。

27. 【参考答案】B。

【解析】线程的实现模型中,根据内核线程和用户线程的对应关系,有三种方式:

(1) 一对一模型,一个内核线程和一个用户线程对应,这种方式执行实现简单,效率最高,但是内核线程较多,开销较大;即使一个用户线程被阻塞,其他用户线程也可以继续运行。

(2) 一对多模型,多个用户线程对应一个内核级线程,这种方式执行内核线程较少,开销较小,但是当用户线程被阻塞时,整个进程都将被阻塞,因此 B 选项入选。

(3) 多对多模型,上述二者的折中方式。

28. 【参考答案】C。

【解析】当物理块是 3 时,页面执行过程是:

访问串	1	3	2	1	1	3	5	1	3	2	1	5
内存	1	1	1	1	1	1	1	1	1	1	1	1
			3	3	3	3	3	3	3	3	3	5
				2	2	2	5	5	5	2	2	2
缺页	√	√	√				√			√		√

缺页次数是 6 次,缺页率是 50%。

当物理块是 4 时,页面执行过程是:

访问串	1	3	2	1	1	3	5	1	3	2	1	5
内存	1	1	1	1	1	1	1	1	1	1	1	1
		3	3	3	3	3	3	3	3	3	3	3
			2	2	2	2	2	2	2	2	2	2
							5	5	5	5	5	5
缺页	√	√	√				√					

缺页次数是 3 次,缺页率是 33%。

29. 【参考答案】D。

【解析】凡是事先进行分区划分的内容管理方式,均会产生内部碎片,其中分页虚拟存储器管理和段页式管理方式都是基于页式管理方式,所以会有内部碎片。固定式分区管理也会事先对内存进行分区,所以三者都会产生内部碎片。

30. 【参考答案】D。

【解析】文件目录项由 16 个磁盘块组成,每个磁盘块也可以作为 1 级间接索引指向 512 个磁盘块,每个磁盘块大小为 1024 字节,因此文件大小是  $16 * 512 * 1024 = 2^{23}B$ 。

31. 【参考答案】C。

【解析】采用 SCAN 算法,且沿着磁道变大的方向移动(外侧是 0,其向内侧移动),于是调度的顺



序是 58 → 130 → 180 → 199 → 42 → 15。于是移动的磁道是  $199 - 58 + 199 - 15 = 325$ 。

32. 【参考答案】B。

【解析】这类题只要记住，时间换空间就是空间变大了，空间换时间就是时间缩短了，一般就不会做错了。总结如下：

各种虚拟存储技术都是时间换空间的技术，包括请求分页、请求分段、请求段页式，这些都是让访问时间增加了，但是扩充了主存的逻辑容量，使得大于主存容量的程序也可以得到执行。如果空间换时间，则各类的缓冲区、缓冲池都是，本来需要在速度很慢的设备上 I/O 的，但是自从划分了些存储区域做缓冲，那么就可以减少访问时间。SPOOLing 技术需有高速大容量且可随机存取的外存支持，通过预输入及缓输出来减少 CPU 等待慢速设备的时间，这是典型的以空间换时间策略的实例。

所以本题选 B。

33. 【参考答案】D。

【解析】物理层的功能是透明传输比特流，那么就需要识别编码信息，既要识别 0 和 1 的编码，又要知道每个信号持续的时间长度，同时还需要决定信号的方向。当然了物理层也要屏蔽物理介质的差异。而 D 选项，避免发送速度过快属于流量控制和拥塞控制，物理层不能进行流量控制和拥塞控制。

34. 【参考答案】D。

【解析】本题考查香农定理的应用。在一条带宽为  $W$  Hz、信噪比为  $S/N$  的有噪声信道的最大数据传输率  $V_{\max}$  为  $W \log_2(1+S/N)$  bps。

先计算信噪比  $S/N$ ：由  $30\text{dB} = 10 \log_{10} S/N$ ，得  $\log_{10} S/N = 3$ ，所以  $S/N = 10^3 = 1000$ 。

$V_{\max} = W \log_2(1+S/N) \text{ bps} = 4000 \log_2(1+1000) \text{ bps} \approx 4000 \times 9.97 \text{ bps} < 40 \text{ kbps}$ 。

35. 【参考答案】A。

【解析】本题考查了有关 GBN 协议的相关机制问题。在 GBN 协议中，接收窗口尺寸被定为 1，从而保证了按序接收数据帧。如果接收窗口内的序号为 4，那么此时接收方需要接收到的帧即为 4 号帧，即便此时接收到正确的 5 号帧，接收端也会自动丢弃该帧从而保证按序接收数据帧。

**注意：**GBN 协议中接收端是没有缓存的，所以也不存在将 5 号帧缓存下来的说法。

36. 【参考答案】C。

【解析】本题考查最小帧长与信道利用率。在确认帧长度和处理时间均可忽略不计的情况下，信道的利用率  $\approx t_{\text{发送时间}} / (t_{\text{发送时间}} + 2t_{\text{传播时间}})$ 。根据信道利用率的计算公式，当发送一帧的时间等于信道的传播时延的 2 倍时，信道利用率是 50%，或者说当发送一帧的时间等于来回路程的传播时延时，效率将是 50%，即  $20\text{ms} \times 2 = 40\text{ms}$ 。现在发送速率是 4000bps，即发送一位需要 0.25ms，则帧长  $40 / 0.25 = 160\text{bit}$ 。



37. 【参考答案】A。

【解析】网络号中第几个字段不相同,就把第几个字段按二进制展开,结果如下:

100.100.00000000.0/18

100.100.01000000.0/18

100.100.10000000.0/18

100.100.11000000.0/18

很明显从第三个字段的第一位开始就已经不同,按照 CIDR 的规则,找到最大能涵盖这四个网络的网络号,故超网的网络号是 100.100.0.0/16

38. 【参考答案】C。

【解析】ARP 协议解决的是一个局域网的 IP 地址到 MAC 地址的映射,他的映射有 4 种情况,分别是,

1) 发送方是主机,把 IP 数据包发送到本网络上的另一个主机。用 ARP 找到目的主机的硬件 MAC 地址。

2) 发送方是主机,要把 IP 数据报发送到另一个网络上的主机。这时用 ARP 找到本网络上的一个路由器(网关)的硬件 MAC 地址。剩下的工作由这个路由器来完成。

3) 发送方是路由器,要把 IP 数据报转发到本网络上的一个主机。用 ARP 找到目的主机的硬件 MAC 地址

4) 发送方是路由器,要把 IP 数据报转发到另一个网络的一个主机。这时用 ARP 找到本网络上的一个路由器(网关)的硬件地址。剩下的工作有这个路由器来完成。所以以太网帧的目的 MAC 地址 R1 的 MAC 地址。同时因为没有 NAT 转换,所以 IP 地址是不变的。

39. 【参考答案】B。

【解析】在慢启动和拥塞避免算法中,拥塞窗口初始值为 1,窗口太小开始按指数增长。当拥塞窗口大于慢启动门限后,停止使用慢启动算法,改用拥塞避免算法。此时,慢启动的门限值初始为 8,当拥塞窗口增大到 8 时改用拥塞避免算法,窗口大小按线性增长,每次增长 1 个报文段。当增加到 12 时,出现超时,重新设置门限值为 6(12 的一半),拥塞窗口再重新设为 1,执行慢启动算法,到门限值为 6 时执行拥塞避免算法。按照上面的算法,拥塞窗口的变化为 1、2、4、8、9、10、11、12、1、2、4、6、7、8、9...,从该序列可以看出,第 12 次传输时拥塞窗口大小为 6。

40. 【参考答案】A。

【解析】本题考查多种协议所在的层数。

链路层: PPP、HDLC、CSMA

网络层: ARP、ICMP、IP、OSPF

传输层: TCP、UDP

应用层: DHCP、RIP、BGP、DNS、FTP、POP3、SMTP、HTTP、MIME

故选 A。

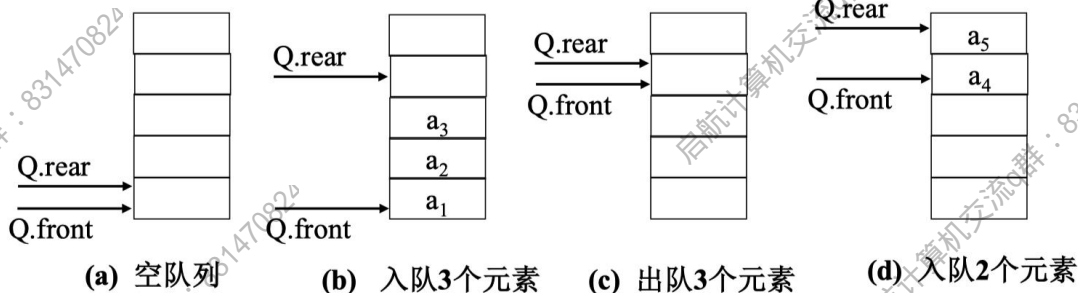




二、综合应用题：41~47 小题，共 70 分。请将答案写在答题纸指定位置上。

41、【答案解析】

(1) 顺序队列中存在“假溢出”现象。因为在入队和出队操作中，头、尾指针只增加不减小，致使被删除元素的空间永远无法重新利用。因此，尽管队列中实际元素个数可能远远小于数组大小，但可能由于尾指针已超出向量空间的上界而不能做入队操作。该现象称为假溢出。如下图所示是数组大小为 5 的顺序队列中队首、队尾指针和队列中元素的变化情况。



(2) 解决办法：可以采用循环队列。

(3) 队空条件： $\text{front} = \text{rear}$ 。

队满条件： $\text{front} = (\text{rear} + 1) \% m$

队长： $(\text{rear} - \text{front} + m) \% m$

PS：入队操作： $\text{rear} = (\text{rear} + 1) \% m$

42. 【答案解析】

解法 1：

(1) 算法的基本设计思想：

注意到旋转之后的数组实际上可以划分成两个排序的子数组，且前面的子数组的元素都大于等于后面子数组的元素，而最小的元素刚好是这两个子数组的分界线。

我们试着用二元查找的思路寻找这个最小的元素。定义两个指针，分别指向数组的第一个元素和最后一个元素。按照题目旋转的规则，第一个元素应该是大于等于最后一个元素的。再定义一个指针指向中间的元素，如果该中间元素位于前面的递增子数组，那么它应大于等于第一个指针指向的元素，此时最小的元素位于右子数组，然后把第一指针指向该中间元素，这样可以在缩小的范围内继续寻找。同样，如果该中间元素位于后面的递增子数组，那么思路和上面是类似的。按照上述思路，第一个指针总是指向前面递增数组的元素，而第二个指针总是指向后面递增数组的元素。最后，第一个指针将指向前面子数组的最后一个元素，而第二个指针会指向后面子数组的第一个元素，此时两个指针相邻，而第二个指针指向的正好是最小元素。这就是循环结束的条件。



扫描二维码,进  
研友交流社区  
获取一手资讯

(2) 算法的实现如下:

```
int Min(int *numbers, int length) {
    if (numbers == 0 || length <= 0)
        return 0;
    int index1 = 0;           // 第一个指针
    int index2 = length - 1;  // 第二个指针
    int indexMid = index1;     // 中间指针
    while (numbers[index1] >= numbers[index2]) {
        if (index2 - index1 == 1) {
            indexMid = index2;
            break;
        }
        indexMid = (index1 + index2) / 2;
        if (numbers[indexMid] >= numbers[index1]) // 在右区间
            index1 = indexMid;
        else if (numbers[indexMid] <= numbers[index2]) // 在左区间
            index2 = indexMid;
    }
    return numbers[indexMid];
}
```

每次都把寻找的范围缩小了一半, 时间复杂度为  $O(\log_2 N)$ 、空间复杂度为  $O(1)$ 。

解法 2: 本题最直观的解法并不难。从头到尾遍历数组一次, 就能找出最小元素, 时间复杂度显然是  $O(N)$ 。但这个思路没有利用输入数组的特性。

#### 43、【答案解析】

(1) 数组  $x$  和  $y$  都按顺序访问, 空间局部性都较好, 但每个数组元素都只被访问一次, 故没有时间局部性。

(2) Cache 共有  $32B/16B=2$  行; 4 个数组元素占一个主存块 (现在的计算机中 float 型一般为 32 位, 占 4B); 数组  $x$  的 8 个元素 (共 32B) 分别存放在主存 40H 开始的 32 个单元中, 共占有 2 个主存块, 其中  $x[0] \sim x[3]$  在第 4 块 (00H~0FH 为第 0 块, 10H~1FH 为第 1 块, 以此类推, 40H~4FH 为第 4 块, 下同),  $x[4] \sim x[7]$  在第 5 块中; 数组  $y$  的 8 个元素分别在主存第 6 块和第 7 块中。所以,  $x[0] \sim x[3]$  和  $y[0] \sim y[3]$  都映射到 Cache 第 0 行;  $x[4] \sim x[7]$  和  $y[4] \sim y[7]$  都映射到 Cache 第 1 行, 如下表所示。因为  $x[i]$  和  $y[i]$  ( $0 \leq i < 8$ ) 总是映射到同一个 Cache 行, 相互淘汰对方, 故每次都不命中, 命中率为 0。

Cache-主存地址	40H~5FH	60H~7FH
第 0 行	$x[0] \sim x[3]$ (第四块)	$y[0] \sim y[3]$ (第六块)
第 1 行	$x[4] \sim x[7]$ (第五块)	$y[4] \sim y[7]$ (第七块)

(3) 若 Cache 改用 2-路组相联, 块大小改为 8B, 则 Cache 共有 4 行, 每组 2 行, 共 2 组。两个数组元素占一个主存块。数组  $x$  占 4 个主存块, 数组元素  $x[0] \sim x[1]$ 、 $x[2] \sim x[3]$ 、 $x[4] \sim x[5]$ 、 $x[6] \sim x[7]$  分别在第 8~11 块中 (与上题同理, 这里 00H~07H 为第 0 块, 08H~0FH 为第 1 块, 以此类推); 数组  $y$  占 4 个主存块, 数组元素  $y[0] \sim y[1]$ 、 $y[2] \sim y[3]$ 、 $y[4] \sim y[5]$ 、 $y[6] \sim y[7]$  分别在第 12~15 块中, 映射关系如下表所示; 因为每组有两行, 所以  $x[i]$  和  $y[i]$  ( $0 \leq i < 8$ ) 虽然映射到同一个 Cache 组, 但



可以存放到同一组的不同 Cache 行内, 因此不会发生冲突。每调入一个主存块, 装入的 2 个数组元素中, 第 2 个数组元素总是命中, 故命中率为 50%。

Cache——主存地址	40H~4FH	50H~5FH	60H~6FH	70H~7FH
第一组	x[0]~x[1]	x[4]~x[5]	y[0]~y[1]	y[4]~y[5]
第二组	x[2]~x[3]	x[6]~x[7]	y[2]~y[3]	y[6]~y[7]

(4) 将数组 x 定义为 12 个元素, 则 x 共有 48B, 使得 y 从主存第 7 块开始存放, 即 x[0]~x[3] 在第 4 块, x[4]~x[7] 在第 5 块, x[8]~x[11] 在第 6 块中; y[0]~y[3] 在第 7 块, y[4]~y[7] 在第 8 块。因而, x[i] 和 y[i] (0≤i≤7) 就不会映射到同一个 Cache 行中, 映射关系如下表所示。每调入一个主存块, 装入 4 个数组元素, 第一个元素不命中, 后面 3 个总命中, 故命中率为 75%。

Cache-主存地址	40H~5FH	60H~7FH	80H~8FH
第 0 行	x[0]~x[3] (第四块)	x[8]~x[11] (第六块)	y[4]~y[7] (第八块)
第 1 行	x[4]~x[7] (第五块)	y[0]~y[3] (第七块)	—

#### 44、【答案解析】

(1) 因为寄存器 R 是 5 位二进制, 所以有  $2^5=32$  个通用寄存器。因为一条指令 32 位, 占 4 个地址, 所以编址单位是  $32\text{bit}/4=8\text{bit}=1$  字节。

(2) 方法 1, 可以根据 C 语言代码判断 (比较简单): 第 1 条指令是给 for 循环中的 i 赋初值, 而之后的指令都是循环执行的, 所以 loop 指向第 2 条指令 (这条指令的含义是根据 i 的值求出相对于 B 首址的偏移), 地址为 00003004H。

方法 2, 根据机器代码判断 (较难): 查看 bne 指令格式, 得到补码 OFFSET=FFF9H, 对应的有符号数真值为 -7, 因为是字偏移量且字长等于指令长度, 所以相当于往前跳转 7 条指令 (PC 在取指完自增后应该从第 9 条指令开始向前跳转 7 条), 即  $8+1-7=2$ , loop 指向第二条指令, 地址为 00003004H。

(3) 循环执行 2~8 号共 7 条指令, 循环 5 次, 其中 5 条运算类指令, 1 条访存类指令, 一条分支跳转类指令。第一条指令是运算类指令, 只执行一次, 整个过程  $7 \times 5 + 1 = 36$  条指令。  
 $\text{CPI} = (5 \times (5 \times 4 + 5 + 3) + 4) / 36 = 4$ 。

$\text{IPS} = \text{主频} / \text{CPI} = 25\text{M}$ ,  $\text{MIPS} = 25$ 。

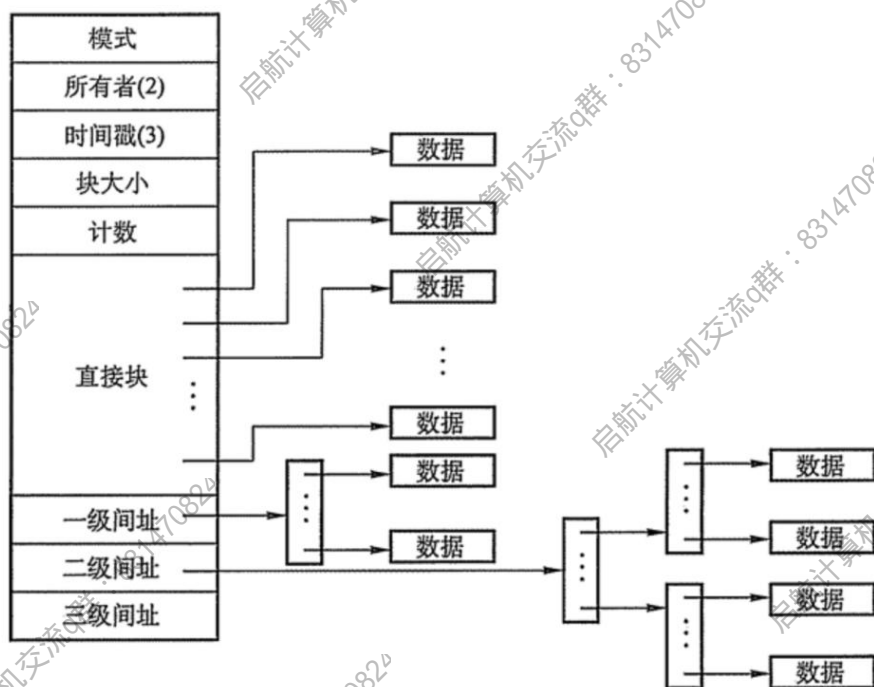
$T = 36 / \text{IPS} = 1.44 \times 10^{-6}\text{s} = 1.44\mu\text{s}$ 。

(4) 指令 1 和 2、2 和 3、3 和 4、4 和 5、6 和 7、7 和 8 都会因为需要使用的寄存器还未被上一条指令写入造成的数据相关产生阻塞。跳转指令会跳回到指令 2 执行, 所以指令 8 和 2 会产生控制相关。



扫描二维码,进入  
启航交流社区  
获取一手资讯

## 45. 【答案解析】



混合索引示意图

(1) 文件大小是  $16 * 1KB + (1KB/4) * 1KB * 2 + (1KB/4)^2 * 1KB$ 。

因为当且除了根目录索引节点在内存外,相关目录和文件数据都不在内存中,且每个目录和索引节点只占一个磁盘块,因此/home/student/course/os/homework 需要访问 5 次磁盘。

(2) 访问 1280 出的记录,因为本文件系统中含有 16 个直接索引,因此直接索引的文件大小是  $16 * 1KB$ ,而  $16KB > 1280 > 1KB$ ,因此只需要访问第二个直接地址就可以。而根目录索引节点已经在内存中,因此访问磁盘的块的个数是 1 次。也即是访问第二个直接地址对应的磁盘块。于是总共访问  $5+1=6$  次。

(3) 访问 1280000 的记录,根据混合索引的存储原理,给定一个文件,首先存储在直接索引中,本题中直接索引大小是  $16KB$ ,接着存储到一次间址,本题中一次间址大小是  $4 * (1KB/4B) * 1KB = 4 * 256 * 1024 = 1\,048\,576$ ,二次间址大小是  $1 * (1KB/4B) * (1KB/4B) * 1KB = 256 * 256 * 1024 = 67\,101\,864$ ,因为  $1\,048\,576 < 1280000 < 67\,101\,864$ ,因此放在二次间址中,需要访问磁盘的次数是 3 次。于是一共是  $5+3$  等于 8 次。

## 46. 【答案解析】

(1) 题目中未说明编址方式时默认按照字节编址, int 占 4 字节,因此  $0x80496dc \sim 0x8049df$  存放了 buf[0]的内容, -259 的十六进制补码为  $0xfffffed$ ;数据存储采用小端方式,因此  $0x80496dc$  的内容是  $0xfd$ ,  $0x80496de$  的内容是  $0xff$ 。

(2) sum 机器代码所占字节数  $= 0x804847d - 0x8048448 + 0x1 = 0x36 = 54$ 。8048455 处的指令是跳转类指令,跳转类指令有可能造成指令流水线的控制阻塞。

(3) 页表大小为 4KB,默认按字节编址,页内地址为低 12 位,即 16 进制的后 3 位,而高 4 位是页号且都为 8048 (十六进制),可以看出 sum 函数存放在同一页中,页号为 8048。





扫描二维码，进  
研友交流社区  
获取一手资讯

## 2023 年全国硕士研究生招生考试 408 计算机超级密押卷三

(4) 调用 `sum` 函数时已将该页调入内存，执行时访问指令不会发生缺页。根据汇编代码分析得到变量 `s` 的地址为 `0x80496f0`，数组 `buf` 首地址为 `0x80496dc`（每循环一次 `+0x4`，共循环 4 次），前 16 位是标记位，因此 `s` 和 `buf` 存放在同一页，且与 `sum` 函数不在同一页，访问 `s` 和 `buf` 一共会产生 1 次缺页中断。

### 47、【答案解析】

(1) 目标地址 `159.226.36.01011011` 与 `R1` 子网掩码 `255.255.255.11100000` 按位与为 `159.226.36.64` 与目的网络地址匹配，所以下一跳 `R0` 直接交付。

(2) 目标地址 `159.226.36.10110011` 与 `255.255.255.11100000` 按位与为 `159.226.36.160` 与前两行 (`R1` 路由表) 不匹配；目标地址 `159.226.36.10110011` 与 `255.255.255.11000000` 按位与为 `159.226.36.128` 与 `R1` 路由表最后一行匹配，所以经 `R1` 的 `I1` 端口传到 `R2` 路由器，再同理由 `R2` 的 `I0` 端口直接交付。