2024 年全国硕士研究生招生考试
计算机学科专业基础试题

（科目代码：408）

考生注意事项

1. 答题前，考生在试题册指定位置上填写考生编号和考生姓名；在答题卡指定位置上填写报考单位、考生姓名和考生编号，并涂写考生编号信息点。
2. 考生须把试题册上的“试卷条形码”粘贴条取下，粘贴在答题卡的“试卷条形码粘贴位置”框中，不按规定粘贴条形码而影响评卷结果的，责任由考生自负。
3. 选择题的答案必须涂写在答题卡和相应题号的选项上，非选择题的答案必须书写在答题卡指定位置的边框区域内，超出答题区域书写的答案无效；在草稿纸、试题册上答题无效。
4. 填（书）写部分必须使用黑色字迹签字笔书写，字迹工整、笔迹清楚；涂写部分必须使用 2B 铅笔涂写。
5. 考试结束，将答题卡和试题册按规定交回。

（以下信息考生必须认真填写）

考生编号															
考生姓名															

AcKing专业IT教育品牌
公众号 · AcKing

一、单项选择题：01 ~ 40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项是最符合题目要求的。

01. 一个带头结点的链表 L，指针 p 指向链表 L 中间的某个结点（非首尾结点）。对于以下代码段：`q = p->next; p->next = q->next; q->next = L->next; L->next = q`，其功能是(　)。

A. 将 p 指向结点移到 q 指向结点后　　B. 将 q 指向结点移到 p 指向结点后

C. 将 p 指向结点插入头结点后　　D. 将 q 指向结点插入头结点后

【答案】D.将q指向节点插入头节点之后

【解析】q=p->next,q是p的下一个节点，p->next=q->next,p的next指向q的下一个的节点，q节点变为自由节点，q->next=L->next,L->next=q，将自由节点q插入到L的下一位。

【知识点参考】AcKing官网（www.acking.com.cn）->算法基础篇->第四章基础数据结构->链表

02. 中缀表达式 `x+y*(z-u)/v` 对应的后缀表达式是(　)。

A. `xyzu-*v/+`　　B. `xuzu-v*+`　　C. `+x/*y-zuv`　　D. `+x*y/-zuv`

【答案】A.xyzu-*v/+

【解析】可以用栈模拟，也可以将其作为表达式树的中序遍历结果，进而还原二叉树，再进行后续遍历，得到后缀表达式。

【知识点参考】AcKing官网（www.acking.com.cn）->算法基础篇->第四章基础数据结构->二叉树

03. p、q、v都是二叉树T中的结点，其中结点v有两个孩子结点，二叉树T的中序遍历为：…，p，v，q，…，则(　)。

A. p 没有右孩子，q 没有左孩子　　B. p 没有右孩子，q 有左孩子

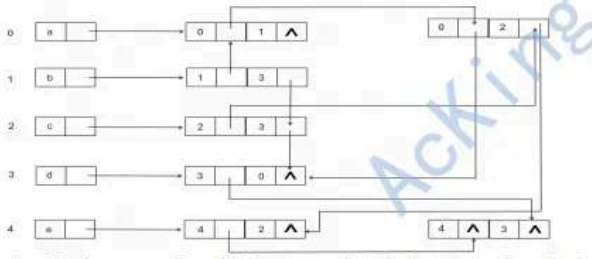
C. p 没有右孩子，q 没有左孩子　　D. p 没有右孩子，q 有左孩子

【答案】A.p没有右孩子，q没有左孩子

【解析】根据中序遍历结果和v是子树的根节点这些信息，则可以判定p，q分别在v的的左右子树上，对于左子树而言，根据中序遍历结果为...p，则p没有右孩子，同理，q没有左孩子。或者假设p有右孩子，q有左孩子，则中序遍历结果中p，v之间一定还有序列，v，q之间也一定还有序列，和题意冲突。

【知识点参考】AcKing官网（www.acking.com.cn）->算法基础篇->第四章基础数据结构->二叉树

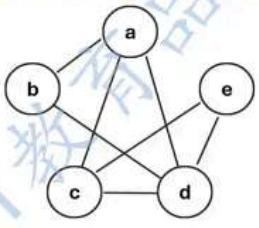
04. 已知如下邻接多重表，请问顶点 b、d 的度分别是(　)。



A. 2,4　　B. 4,2　　C. 2,3　　D. 3,2

【答案】A.2，4

【解析】根据邻接多重表还原图，由下图可知，b的度为2，d的度为4



公众号 · AcKing

05. 以下存储结构中，不适用于折半查找的是(　)。

I. 有序链表　　II. 无序数组　　III. 有序静态链表　　IV. 无序静态链表

A. 仅I和III　　B. 仅II和IV　　C. 仅II、III、IV　　D. I、II、III、IV

【答案】D. I、II、III、IV

【解析】折半查找也叫二分搜索，可以在有序顺序表中应用。

【知识点参考】AcKing官网（www.acking.com.cn）->算法基础篇->第三章基础算法->二分搜索

06. KMP 算法使用修正后的 next 数组进行模式匹配，模式串 s = "aabaab"，主串中某个字符失配时，s 右滑最长距离为(　)。

A. 5　　B. 4　　C. 3　　D. 2

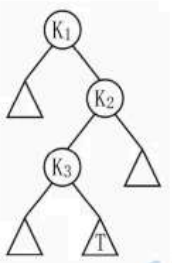
【答案】A. 5

【解析】传统KMP使用next数组，不过会存在j=next[j]的情况，因此采用next_val[]来优化，处理好next[]和next_val[]后，举例即可，比如倒数第二个元素a匹配失败，会回到-1的位置，从-1的位置走到4的位置需要右滑的距离为5位。

下标	0	1	2	3	4	5
	a	a	b	a	a	b
next	-1	0	1	0	1	2
next_val	-1	-1	1	-1	-1	1

【知识点参考】AcKing官网（www.acking.com.cn）->算法提高篇->第二章字符串算法->KMP

07. 一棵二叉搜索树如下图所示，图中 K_1、K_2、K_3 表示结点中保存的关键字，图中三角形表示子树。则图中子树 T 中任意结点保存的关键字 X 满足(　)。



A. $X < K_1$　　B. $X > K_2$　　C. $K_1 < X < K_3$　　D. $K_3 < X < K_2$

【答案】D. $K_3<X<K_2$

【解析】根据二叉搜索树的性质，K_2的左子树的节点关键字均小于K_2，可以得出$K_3<K_2$，T中所有节点小于K_2，同理也可以得出K_3的右子树节点均大于K_3，即T中所有节点>K_3，X是T中节点，由此可以得出$K_3<X<K_2$。

【知识点参考】AcKing官网（www.acking.com.cn）->算法基础篇->第四章基础数据结构->二叉搜索树

08. 使用快速排序算法对含 N 个元素的数组 M 进行排序，若第一趟排序将除枢轴外的 N – 1 个元素划分为 P 和 Q 两个部分，则下列叙述中，正确的是(　)。

A. P 和 Q 块间有序　　B. P 和 Q 均块内有序

C. P 和 Q 的元素个数大致相等　　D. P 和 Q 中均不存在相等的元素

【答案】A. P和Q块间有序

【解析】本题考察快排的原理，快排第一趟找到枢轴之后，枢轴的左边的元素即P块都是小于枢轴的，枢轴的右边的元素即Q块是大于枢轴的，所以P和Q块间有序，然而块内未必是有序的，所以还需要继续对P和Q两块进行递归快排。

【知识点参考】AcKing官网（www.acking.com.cn）->算法进阶篇->第三章排序算法->快速排序

公众号 · AcKing

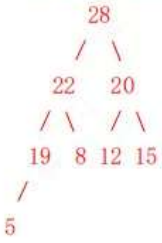
09. 大根堆初始序列为：28, 22, 20, 19, 8, 12, 15, 5，对该堆进行两次删除操作后，得到的新堆是(　)。

A. 20, 19, 15, 12, 8, 5　　B. 20, 19, 15, 5, 8, 12　　C. 20, 19, 12, 15, 8, 5　　D. 20, 19, 8, 12, 15, 5

【答案】B. 20，19，15，5，8，12

【解析】本题考察堆的创建和堆调整，先根据初始序列建堆，然后依次删除堆顶元素。

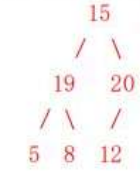
堆的初始形态为：



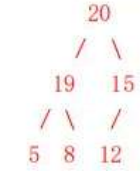
删除28后，堆的形态为：

重新调整为大根堆，形态为：

删除22后，堆的形态为：



重新调整为大根堆，形态为：



【知识点参考】AcKing官网（www.acking.com.cn）->算法进阶篇->第三章排序算法->堆排序

AcKing专业IT教育品牌
公众号 · AcKing

10. 对如下三个升序序列：{3,5}、{7,9}、{6}，进行二路归并排序，关键字的比较次数是(　)。

A. 3　　B. 4　　C. 5　　D. 6

【答案】C.5

【解析】本题考察归并排序的原理，首先{3,5}和{7,9}进行归并，需要比较2次，合并之后序列变为{3，5，7，9}，然后和{6}进行归并，需要比较3次，总比较次数为5次。

【知识点参考】AcKing官网（www.acking.com.cn）->算法进阶篇->第三章排序算法->归并排序

11. 外部排序使用败者树进行升序归并，“冠军”结点保存的是(　)。

A. 最大关键字　　B. 最大关键字所在的归并段号

C. 最小关键字　　D. 最小关键字所在的归并段号

【答案】D.最小关键字所在的归并段号

【解析】

外部排序指的是大文件的排序，即待排序的记录存储在外存储器上，待排序的文件无法一次装入内存，需要在内存和外部存储器之间进行多次数据交换，以达到排序整个文件的目的。

外部排序最常用的算法是多路归并排序，即将原文件分解成多个能够一次性装入内存的部分分别把每一部分调入内存完成排序。然后，对已经排序的子文件进行归并排序。

在外部排序中，利用败者树进行多路归并时，败者树中记录“冠军”的节点保存的是最小关键字所在的归并段号。因为在多路归并排序中，我们总是希望能够快速找到所有归并段中的最小元素，以便将其输出到排序结果中。败者树的设计正是为了实现这个目标：在每一轮比较中，我们总是将最小元素（即“冠军”）留在树根，而将“败者”保存在内存节点中。

12. 执行如下 C 语言代码之后，变量 j 的值是(　)。

```
int i = 32777;
short si = i;
int j = si;
```

A. −32777　　B. −32759　　C. 32759　　D. 32777

【答案】B.-32759

【解析】int i=32777,赋值给short后，由于short只有16位，则只会保留i的低16位，同时，short的上限为2^{15}-1即32767,所以32777在short类型的变量中会出现数据溢出问题，32777=32767+9，则溢出后的二进制补码为1000 0000 0000 1001。再将此数转为int，此时将进行符号扩展，补高16位的时候取决于当前short的符号位，若符号位为1，则高16位补16个1，若符号位为0，则高16位补16个0，所以本题补16个1，结果为补码1111 .... 1000 0000 0000 1001，换算成原码再计算，选B。或者使用排除法，最终通过补码的最高位1可以断定该数为负数，A，B里面选，32777对于short已经溢出，转为short再符号扩展，并不只是单纯地改变了符号，故排除A，选B。

【知识点参考】AcKing官网（www.acking.com.cn）->C++语法篇->第二章基本数据类型->数据类型、原码、反码、补码

13. 伪指令指汇编语言程序中实现特定功能的指令序列。下列选项中，CPU 能理解并直接执行的是(　)。

I. 伪指令　　II. 微指令　　III. 机器指令　　IV. 汇编指令

A. 仅 I 和 IV　　B. 仅 II 和 III　　C. 仅 III 和 IV　　D. 仅 I、III 和 IV

【答案】B.仅II和III

【解析】伪指令：伪指令不是真正的CPU指令，而是由汇编器识别和执行的一些特殊指令。

微指令：微指令是计算机硬件级别指令，一条机器指令所完成的操作划分成若干条微指令来完成。

机器指令：CPU可以直接解析和执行的二进制代码。

汇编指令：低级别的编程语言，与机器指令一一对应，不过需要汇编器转换为机器指令，CPU才能执行。

公众号 · AcKing

14. 某科学实验中，需要使用大量的整型参数，为了在保证数据精度的基础上提高运算速度，需要选择合理的数据表示方法。若整型参数 α、β 的取值范围分别为：$-2^{-20} \sim 2^{20}$、$-2^{-40} \sim 2^{40}$，则下列选项中，α 和 β 最适宜采用的数据表示方法分别是(　)。

A. 32 位整数、32 位整数　　B. 单精度浮点数、单精度浮点数

C. 32 位整数、双精度浮点数　　D. 单精度浮点数、双精度浮点数

【答案】C.32位整数、双精度浮点数

【解析】α最适宜采用32位整数，32位整数可以表示-2^{31}~$2^{31}-1$的范围，对于α的取值范围，可以用32位整数表示，不会发生溢出或者精度损失，而且整数运算比浮点数运算更快，更省空间，所以优先使用32位整数而不是单精度浮点数。排除B、D，根据β的取值范围，可以使用双精度浮点数存储，双精度浮点数存储范围为-2^{1024}~2^{1024}，不会发生溢出问题，双精度浮点的小数精度为15-16位，比起单精度浮点，基本上不会发生精度损失。

【知识点参考】AcKing官网（www.acking.com.cn）->C++语法篇->第二章基本数据类型->数据类型

15. 下列关于整数乘法运算的叙述中，错误的是(　)。

A. 用阵列乘法器实现乘运算可以在一个时钟周期完成

B. 用 ALU 和移位器实现的乘运算无法在一个时钟周期内完成

C. 变量与常数的乘运算可编译优化为若干天傲移位及加减运算指令

D. 两个变量的乘运算无法编译为移位及加法等指令的循环实现

【答案】D.两个变量的乘运算无法编译为移位及加法等指令的循环实现

【解析】两个变量的乘运算可以编译为移位及加法等指令的循环实现，这种方法称为移位-加法乘法算法，它是一种基于二进制的乘法算法，可以用来计算两个变量的乘积。

16. 对于页式虚拟存储管理系统，下列关于存储器层次结构的叙述中，错误的是(　)。

A. Cache-主存层次的交换单位为主存块，主存-外存冲刺的交换单位为页

B. Cache-主存层次替换算法由硬件实现，主存-外存层次由软件实现

C. Cache-主存层次可采用回写法写策略，主存-外存层次通常采用回写法

D. Cache-主存层次可采用直接映射，主存-外存层次通常采用直接映射

【答案】D.Cache-主存层次可采用直接映射，主存-外存层次通常采用直接映射

【解析】主存-外存层次的映射方式通常采用页式映射，而不是直接映射。页式映射是一种将虚拟地址空间分割为固定大小的单元（页）的映射方式，它将页映射到物理地址的空间的相应单元（页框）中。页式映射的优点是减少了内外存之间的交换次数，提高了空间利用率，缺点是增加了地址转换的开销，需要维护页表。

17. 某计算机按字节编址，采用页式虚拟存储管理方式，虚拟地址为 32 位，主存地址为 30 位，页大小为 1KB。若 TLB 共有 32 个表项，采用 4 路组相联映射方式，则 TLB 表项中标记字段的位数至少是(　)。

A. 17　　B. 18　　C. 19　　D. 20

【答案】C.19

【解析】根据题意可知，虚拟地址为32位，主存地址为30位，页大小为1KB，因此，虚拟页号为32-10=22位，物理页号为30-10=20位。TLB共有32个表项，采用4路组相联映射方式。因此，TLB共有32/4=8组，每组有4路。索引字段为$\log_2 8$=3位。标记字段位数=虚拟页号位数-索引字段位数=22-3=19位。

公众号 · AcKing

18. 下列事件中，不是在 MMU 地址转换过程中检测的是(　)。

A. 访问越权　　B. Cache 缺失　　C. 页面缺失　　D. TLB 缺失

【答案】B. Cache缺失

【解析】MMU地址转换过程中，会检测以下时间：访问越权、页面缺失、TLB缺失。Cache缺失不是在MMU地址转换过程中检测的，而是在CPU访问主存数据检测的。

19. 5 段流水线 RISC 说法错误的是(　)。

A. 选项略...

B. 选项略...

C. 所有数据冒险都可以通过加入转发（旁路）电路解决

D. 所有数据相关都可以通过添加nop指令以及调整指令顺序来解决

【答案】C. 所有数据冒险都可以通过加入转发（旁路）电路解决

【解析】5段流水线RISC是一种将指令执行过程分为五个阶段的流水线结构，这五个阶段分别是取指（IF）、译码（ID）、执行（EX）、访存（MEM）和写回（WB）。

数据冒险是指在流水线中，一个指令需要使用前面一个或多个指令的执行结果作为操作数，但是这些结果还没有准备好，导致流水线暂停或错误的情况。

解决数据冒险的方法有几种：转发或旁路、插入空操作或气泡、调整指令顺序。

20. 存储器总线的时钟频率为 420MHz，总线宽度为 64 位，每个时钟周期传送 2 次数据，支持突发传输，最多传 8 次，第一个时钟传地址和读写命令，从第 4 ~ 7 个始终连续传 8 次。总线带宽最大传输速率是(　)。

A. 3.84GB/s　　B. 6.72GB/s　　C. 30.72GB/s　　D. 53.76GB/s

【答案】A. 3.84GB/S

【解析】时钟频率位120MHz，总线宽度位64位即8B，第一个时钟传地址和读写命令，从第4~7个始终连续传8次，实际上就是7个时钟周期传输8次数据，共8B*8=64B，总线带宽最大传输速率为：420MHz*64B/7=3.84GB/s

21. 下列关于 I/O 控制方式的叙述中，错误的是(　)。

A. 中断屏蔽字决定中断响应顺序

B. 选项略...

C. 保存通用寄存器和设置新中断屏蔽字由软件实现

D. 单重中断方式下，中断处理时CPU处于关中断状态

【答案】A. 中断屏蔽字决定中断响应顺序

【解析】中断是指外部设备或者程序需要CPU的服务时，向CPU发出一个信号，请求CPU暂停当前的任务，转而处理中断请求的情况。

中断响应顺序是指当同时有多个中断请求时，CPU按照一定的优先级顺序来选择响应哪个中断的规则。

中断屏蔽字只能控制中断的开关，而不能控制中断的顺序，中断的顺序由中断的优先级决定，优先级高的中断先被响应，优先级低的中断后被响应。

公众号 · AcKing

22. 在 DMA 控制方式中，DMA 控制器控制的数据传输通路位于(　)。

A. 选项略...

B. 选项略...

C. 设备接口和主存之间

D. 设备接口和DMA控制器之间

【答案】C. 设备接口和主存之间

【解析】在DMA方式中，DMA控制器控制的数据传输通路位于设备接口和主存之间。这是因为DMA（Direct Memory Access，直接存储器访问）的主要功能是在外设和存储器之间或者存储器和存储器之间提供高速数据传输。

23. 下面关于中断和异常的说法中，错误的是(　)。

A. 中断或异常发生时，CPU处于内核态

B. 选项略...

C. 选项略...

D. 中断处理服务程序发生时，CPU处于内核态

【答案】C. 设备接口和主存之间

【解析】在DMA方式中，DMA控制器控制的数据传输通路位于设备接口和主存之间。这是因为DMA（Direct Memory Access，直接存储器访问）的主要功能是在外设和存储器之间或者存储器和存储器之间提供高速数据传输。

24. 终止进程时，不一定执行的是(　)。

A. 终止子进程　　B. 回收分配的内存资源

C. 撤销进程PCB　　D. 回收进程占用的设备

【答案】A. 终止子进程

【解析】当用户终止进程时，不一定终止子进程。因为子进程的生命周期并不总是与父进程紧密相关联，在某些情况下，即使父进程被终止，子进程也可以继续运行，如孤儿进程和僵尸进程。

25. 支持页式存储管理的系统，进程切换时 OS 要执行(　)。

I. 更新 PC 值　　II. 更新栈基址寄存器值（ebp）　　III. 更新页表基址寄存器值

A. 仅III　　B. 仅I、II　　C. 仅I、III　　D. I、II、III

【答案】D. I、II、III

【解析】I. 更新程序计数器的值：程序计数器存储了下一条要执行的指令的地址。当进程切换时，操作系统需要更新程序计数器的值，以便于新的进程能从正确的位置开始执行。

II. 更新栈基址寄存器的值：栈基址寄存器存储了当前进程栈的基址。当进程切换时，操作系统需要更新栈基址寄存器的值，以确保新的进程使用正确的栈。

III. 更新页表基址寄存器值：页基址寄存器存储了当前进程的页表基址。当进程切换时，操作系统需要更新页基址寄存器的值，以确保新的进程能正确地访问其内存空间。

公众号 · AcKing

26. 文件系统需要额外的外存空间记录空闲块的位置，占用外存空间大小与当前空闲块数量无关的是(　)。

A. 位图法　　B. 空闲表　　C. 成组链接　　D. 空闲链表

【答案】A. 位图法

【解析】A. 文件系统需要额外的外存空间记录空闲块的位置，占用外存空间大小与当前空闲块数量无关的是位示图。位示图是一种常用的记录空闲块位置的方法，它使用一个位来表示一个块是否空闲。位示图的大小取决于磁盘的总块数，而与当前的空闲块数量无关。

B. 空闲表:空闲表是一种记录磁盘空闲块位置的方法，它使用一个表来记录空闲块的位置。空闲表的大小会随着空闲块的数量的变化而变化。

C. 成组链接：成组链接是一种记录该组中其他块的位置。成组链接的大小会随着空闲块数量的变化而变化。

D. 空闲链表：空闲链表是一种记录磁盘空闲块位置的方法，它使用一个链表来记录所有空闲块的位置。空闲链表的大小会随着空闲块数量的变化而变化。

27. 回收分区时，仅合并大小相等的空闲分区的算法是(　)。

A. 伙伴算法　　B. 最佳适应算法　　C. 最坏适应算法　　D. 首次适应算法

【答案】A. 位图法

【解析】A. 伙伴算法是一种特殊的内存分配算法，他在分配和回收内存时，只合并大小相等的空闲分区。这种算法的优点是简单且执行速度快，但可能会导致内存碎片；

B. 最佳适应算法：它在分配内存时，会选择大小最接近所需的空闲分区。这种算法的优点是可以减少内存的浪费，但可能会导致大量的小碎片。

C. 最坏适应算法：它在分配内存时，会选择最大的空闲分区。这种算法的优点是可以减少内存的碎片，但可能会导致大量的大碎片。

D. 首次适应算法：它在分配内存时，会选择第一个满足所需的空闲分区。这种算法的优点是简单且执行速度快，但可能会导致内存的碎片。

28. 进程 P 有一个线程 T，打开文件后获得 fd，再创建线程 Ta、Tb，则线程 Ta、Tb 可共享的资源是(　)。

I. 进程 P 的地址空间　　II. 线程 T 的栈　　III. fd

A. 仅I　　B. 仅I、III　　C. 仅II、III　　D. I、II、III

【答案】B. 仅I、III

【解析】I. 进程P的地址空间：在同一进程中的所有线程共享该进程的地址空间。这意味着，线程Ta和Tb可以访问进程P的全局变量，因为这些变量存储在进程的地址空间中。此外，如果线程T在堆上分配了内存，那么线程Ta和Tb也可以访问这些内存，因为堆是存储在进程的地址空间中的。

II. 线程T的栈：每个线程都有自己的栈，这是线程的私有资源，不会被其他线程共享。栈用于存储函数调用的局部变量和返回地址。由于每个线程可能有不同的函数调用序列，因此每个线程需要由自己的栈，因此，线程Ta和Tb不能访问线程T的栈。

III. 文件描述符fd：在同一进程中的所有线程共享该进程打开的文件描述符。文件描述符是一个整数，用于表示进程打开的文件。当线程T打开一个文件时，操作系统会返回一个文件描述符，然后线程T、Ta和Tb都可以使用这个文件描述符来读写该文件，这是因为，尽管每个线程有自己的栈，但是它们共享其余的进程资源，包括文件描述符。

公众号 · AcKing

29. 包含文件按名查找功能的系统调用是(　)。

A. open()　　B. read()　　C. write()　　D. close()

【答案】A.open()

【解析】A.open()系统调用用于打开一个文件。它需要一个文件名作为参数，因此它包含了按名查找文件的功能。如果文件存在并且进程有足够的权限，open()会成功打开文件并返回一个文件描述符，否则，它会返回一个错误。

B.read()系统调用用于从已打开的文件中读取数据。
C.write()系统调用用于向已打开的文件中写入数据。
D.close()系统调用用于关闭已打开的文件。

30. 系统采用时间片轮转调度，时间片为5ms，有10个进程，初始状态均处于就绪队列，执行结束前仅处于执行态或就绪态，队尾进程P所需CPU时间最短，为25ms，不考虑系统开销，则P的周转时间为(　)。

A. 200ms　　B. 205ms　　C. 250ms　　D. 295ms

【答案】C.250ms

【解析】由于使用的是轮转调度算法，进程P在每次执行一个时间片后，都需要重新回到就绪队列的末尾等待下一次的时间片。所以，实际上，进程P的每一个时间片之间都有一个完整的轮转周期的等待时间：10*5ms=50ms，进程P需要执行25/5个时间片，所有中间有4个完整的轮转周期再加上P的周转时间为，总共需要5个轮转周期：5*50ms=250ms 。

31. 键盘中断服务例程执行结束时，所输入的数据存放位置是(　)。

A. 用户缓冲区　　B. CPU的通用寄存器

C. 内核缓冲区　　D. 键盘控制器的数据缓冲区

【答案】C.内核缓冲区

【解析】键盘中断服务例程执行结束时，所输入的数据存放位置是内核缓冲区，当键盘中断发生时，操作系统会暂停当前正在执行的任务，转而执行键盘中断服务例程。这个例程会从键盘控制器读取输入的数据，并将其存放在内核缓冲区中，然后当用户程序准备好读取数据时，它可以从内核缓冲区中获取。

32. 磁盘调度算法是CSCAN，磁道号是0～399，完成200号磁道请求后，磁头向磁道号变小的方向移动，此时有7个请求：300, 120, 110, 0, 160, 210, 399。完成上述访问请求后，磁头移动距离是(　)。

A. 599　　B. 619　　C. 788　　D. 799

【答案】C.788

【解析】在CSCAN中，磁头会在一个方向上移动，直到达到磁道的一端，然后立即返回到另一端，再次开始扫描。

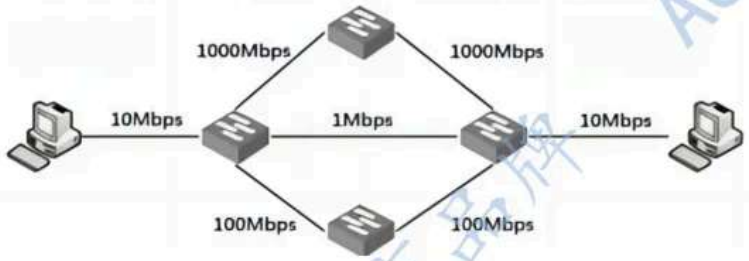
首先磁头会移动到160号磁道，然后依次是120号、110号、0号，接着磁头移动到开头，然后向磁道号减少的方向移动，依次移动到399号、300号、210号，完成所有请求后，磁头移动的总距离为：

200-160=40；　160-110=50；　110-0=110；　399-0=399；　399-300=90；　300-210=90；

磁头移动的总距离为40+50+110+399+99+90=788

公众号 · AcKing

33. 若分组交换网络及每段链路的带宽如下图所示，则 H1 到 H2 的最大吞吐量约为(　)。



A. 1Mbps　　B. 10Mbps　　C. 100Mbps　　D. 1000Mbps

【答案】B. 10Mbps

【解析】H1和H2两端的最大传输速率为10Mbps，虽然中间的路由器最大传输速率为1000Mbps，但是根据网络最大流算法，H1和H2的最大吞吐量等价于最大流，最大流为10Mbps

34. 在下列二进制数字调制方法中，需要 2 个不同频率载波的是(　)。

A. ASP　　B. PSK　　C. FSK　　D. DPSK

【答案】C. FSK

【解析】数字调制方法中没有ASP。

B. PSK：相位移位键控是一种数字调制计数，它通过改变载波信号的相位来表示二进制数据。在这种调制方式中，只需要一个固定频率的载波。

C. FSK：频率移位键控，是一种数字调制计数，它通过改变载波信号的频率来表示二进制数据。在这种调制方式中，需要两个不同频率的载波，一个表示二进制0，另一个表示二进制1。

D. DPSK：差分相位移位键控是一种相位调制技术。

35. 若UDP协议在计算校验和过程中，计算机得到中间结果为1011 1001 1011 0110时，还需要加上最后一个16位数0110 0101 1100 0101，则最终计算得到的校验和是（　）。

A. 0001 1111 0111 1011　　B. 0001 1111 0111 1100

C. 1110 0000 1000 0011　　D. 1110 0000 1000 0100

【答案】C. 1110 0000 1000 0011

【解析】UDP是一种传输层协议，UDP校验和计算的步骤如下：

将UDP首部和UDP数据字段合并成一个伪首部和一个伪数据字段。伪首部的内容包括源IP地址、目的IP地址、协议类型、UDP数据长度。伪数据字段的内容为UDP首部和数据字段本身。

对伪首部和伪数据字段进行16位的字节级求和，得到一个结果。

对结果进行反码运算，得到最终的校验和。

1011 1001 1011 0110

+0110 0101 1100 0101

————————————————

10001 1111 0111 1011

最高位产生进位所以需要在末尾加1，可得：0001 1111 0111 1100

取反可得：1110 0000 1000 0011

公众号 · AcKing

36. 在采用CSMA/CA的802.11无线局域网中，DIFS=128μs,SIFS=28μs,RTS、CTS和ACK帧的传输时延分别是3μs、2μs和2μs,忽略信号传播时延，若主机A欲向AP发送一个总长度为1998B的数据帧，无线链路带宽为54Mbps,则隐藏站B到AP发送的CTS帧时，设置的网络分配向量NAV的值是(　　)。

A.326μs　　B.354μs　　C.385μs　　D.513μs

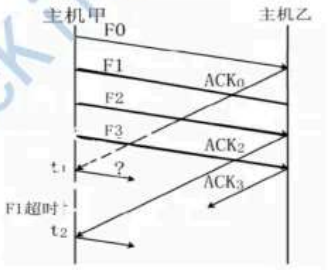
【答案】B. 354

【解析】在CSMA/CA的802. 11无线局域网中，网路分配向量（NAV）是用来告诉其他节点预计要占用无线媒体多长时间的一种机制。在发送RTS（请求发送）和CTS（清除发送）帧时，发送节点会在帧头中包含一个持续时间字段，这个字段表示从发送当前帧到接收到最后一个ACK帧所需的时间。其他节点在接收到RTS或CTS帧后，会根据这个持续时间字段设置自己的NAV，从而避免在这段时间内发送数据，防止发生冲突。

NAV的时间就是其他节点推迟访问的时间，约等于SIFS时间+数据发送时间++SIFS时间+ACK帧传播时延。

根据题目中的信息，我们可以计算出设置NAV的值。首先，我们需要计算发送数据帧所需的时间。数据帧的总长度为1998B，无线链路带宽为54Mbps，所以发送数据帧所需的时间为：1998B/54Mbps=296μs,则总时间=28μs+296μs+28μs+2μs=354μs

37. 主机甲通过选择重传（SR）滑动窗口协议向主机乙发送帧的部分过程如下图所示。F为数据帧，ACKx为确认帧，X是位数为了比特的序号。乙只对正确接收的数据帧进行独立确认。发送窗口与接收窗口大小相同且均为最大值。甲在 t_1 时刻和 t_2 时刻发送的数据帧分别是(　　)。



A. F1、F3　　B. F1、F4　　C. F3、F1　　D. F4、F1

【答案】D. F4、F1

【解析】在t1时刻主机甲接收到了主机乙发送的ACK0，暂时没有出错的帧，所以继续发送下一个数据帧F4；在t2时刻主机甲收到了F1超时的错误，根据选择重传协议，发送方只需要重传出错的帧，而不是所有的帧，所以只需要重新发送F1。

AcKing专业IT教育品牌
公众号 · AcKing

38. 主机 A 向服务器请求 web 页面，该页面由一个 html 文件以及所引用的长度大小为 3MSS 的图像文件构成。最长报文段生存时间为 30ms，rtt 为 0.01ms。请问主机 A 从发送 web 请求到关闭 tcp 连接，所需的最短时间是(　)。

A. 30.03ms　　B. 30.04ms　　C. 60.03ms　　D. 60.04ms

【答案】D. 60.04ms

【解析】在TCP连接中，从建立连接到关闭连接的过程包括以下步骤：

1. 建立链接：这个过程通常为称为三次握手，SYN，SYN-ACK，ACK，需要一个往返时间RTT。

2. 数据传输：主机H向服务器发送长度为3000B的报文。由于最大报文段长度MSS=1000B，第一次拥塞窗口为1发送1000B，第二次拥塞窗口为2，发送2000B，所以一共消耗2个最长报文段寿命，2个RTT。

3. 关闭连接：这个过程通常称为四次挥手，FIN，ACK，FIN，ACK。

第一次挥手和第三次挥手构成一个RTT，第二次和第三次是连续发送的，第四次发出去的时候和客户端等待的2MSL时间重叠，所以需要1个RTT。

所需时间1RTT+2RTT+1RTT+2*MSL=4RTT+2*MSL。

将RTT=10ms=0.01s和MSL=30s代入上式，得到60.04s。

39. 已知如下 3 个 vlan 及其分布图，主机 E 的 ARP 表不可能有的一项是(　)。

A. IP2.168.3.81,　　00-18A2-3B-36-21,　14:32:00

B. IP2.168.3.91,　　00-3E-C2-39-12-B5,　14:37:00

C. IP2.168.3.125,　　00-E5-78-4A-09-B2,　14:35:00

D. IP2.168.3.129,　　00-08-6E-05-A7-82,　14:52:00

【答案】D. IP2.168.3.129,　　00-08-6E-05-A7-82,　　14:52:00

【解析】本题无图，暂时无解析。

40. 若浏览器不支持并行 TCP 连接，使用非持久的 HTTP/1.0 协议请求浏览 1 个 web 页，该页中引用同一个网站上 7 个小图像文件，则从浏览器传输 web 页请求建立 TCP 连接开始后，到接收完所有内容为止。所需要的往返时间 RTT 数至少是(　)。

A. 4　　B. 9　　C. 14　　D. 16

【答案】D. 16

【解析】在HTTP/1.0协议中，如果浏览器不支持并行TCP连接，那么每个请求（包括主页面和每个图像文件）都需要单独建立一个TCP连接。每个TCP连接的建立都需要1个往返时间RTT，因此，对于主页面和7个图像文件的请求，总共需要8个RTT。此外，每个TCP连接在数据传输完成后都需要关闭，每个关闭连接也需要一个RTT，共8个RTT。所以从浏览器为传输Web页请求建立TCP连接开始，到接收完所有内容为止，所需要的往返时间RTT数至少是8（建立连接）+8（关闭连接）=16

IT教育品牌　AcKing专业IT教育品牌　公众号 · AcKing

二、综合应用题：41～47小题，共70分。

41（13分）2023年10月26日，神州十七号载人飞船发射取得圆满成功，再次彰显了中国航天事业的辉煌成就。载人航天工程是包含众多子工程的复杂系统工程，为了保证工程的有序开展，需要明确各子工程的前导工程，以协调各子工程的实施。该问题可以简化、抽象为有向图的拓扑序列问题。已知有向图G采用邻接矩阵存储，类型定义如下：

```
typedef struct                          //图的类型定义
{
   int numVertices,numEdges;            //图的顶点数和有向边数
   char VerticesList[MAXV];             //顶点表，MAXV为已定义常量
   int Edge[MAXV][MAXV];                //邻接矩阵
}MGraph;
```

请设计算法：int uniquely(MGraph G),判定G是否存在唯一的拓扑序列，若是则返回1，否则返回0。

要求：

（1）给出算法的基本设计思想（4分）。

【答案】

初始化：首先，我们需要一个数组来存储每个顶点的入度。我们遍历邻接矩阵，计算每个顶点的入度。

初始一个计数器作为变量用于记录有多少个入度为0的节点

遍历每个节点：首先计算每个顶点的入度，然后在每次迭代中找到并移除了入度为0的顶点。如果在任何时刻存在多个入度为0的顶点，那么就返回0，表示存在多个有效的拓扑序列。如果成功完成了拓扑排序，那么就返回1，表示存在唯一的拓扑序列。

（2）根据设计思想，采用C或C++语言描述算法，关键之处给出注释（9分）。

【答案】

```
int uniquely(MGraph G){
    int inDegree[MAXV]={0};    //初始化入度数组
    //计算每个顶点的入度
    for(int i=0;i<G.numVertices;i++){
        for(int j=0;j<G.numVertices;j++){
            if(G.Edge[i][j]!=0)
                inDegree[j]++;
        }
    }
```

公众号 · AcKing

```
//对每个顶点进行处理
for(int i=0;i<G.numVertices;i++){
    int zeroInDegreeCount=0;      //入度为0的顶点数量
    int zeroInDegreeIndex=-1;     //入度为0的顶点索引
    //找到入度为0的顶点
    for(int j=0;j<G.numVertices;j++){
        if(inDegree[j]==0){
            zeroInDegreeCount++;
            zeroInDegreeIndex=j;
        }
    }
    //如果存在多个入度为0的顶点或者没有找到入度为0的点，则返回0
if (zeroInDegreeCount!=1) return 0;
    //将找到的入度为0的顶点从图中移除
    inDegree[zeroInDegreeIndex]=-1;
    for(int j=0;j<G.numVertices;j++){
        if(G.Edge[zeroInDegreeIndex][j]==1){
            inDegree[i]--;
        }
    }
}
//如果成功完成了拓扑排序，则返回1
return 1;
}
```

42. (10 分) 将关键字 20, 3, 11, 18, 9, 14, 7 依次存储到长度为 11 的散列表 HT 中，散列函数为 H(key) = (key × 3)%11，H0 为初始散列地址，H1、H2、H3、…、Hk 分别为第 1 次冲突、第 2 次冲突、第 3 次冲突、…、第 k 次冲突时探测的地址。$Hk = (H0 + k^2)\%11$。请回答下列问题：

(1) 画出 HT，算装填因子。

(2) 查找关键字 14 时的关键字比较序列。

(3) 查找关键字 8 失败时的哈希地址。

(1)画出HT，算装填因子。

【答案】

散列表HT如下：

散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	11		14	7		20	9			3	18		
冲突次数	1		3	2		1	2			1	1		

装填因子等于散列表中已经被填充的位置的数量除以散列表的总长度因此，本题的填装因子是7/11。

AcKing专业IT教育品牌
公众号 · AcKing

(2)查找关键字14时的关键字比较序列

【答案】

1.首先，我们计算14的散列地址：H（14）=（14*3）%11=42%11=9。

2.我们查看散列表中索引为9的位置，发现哪里存储的是关键字3，此时产生哈希冲突。

3.由于我们使用的是二次探查，所以计算下一个散列地址：H1=（H0+1*1）%11=（9+1）%11=10。发现那哪里存储的是关键字18，再次遇到哈希冲突。

4.继续计算下一个散列地址：H2 =（H0+2*2）%11=(9+4)%11=2，找到关键字14.

(3)查找关键字8失败时的哈希地址。

【答案】

1.计算8的散列地址：H（8）= （8*3）%11 = 24%11=2，索引为2的位置发现关键字18，遇到冲突。

2.使用二次探查，计算下一个散列地址：H1=（H0+1*1）%11=（2+1）%11=3。索引为3的位置存储的是关键字7，遇到冲突。

3.我们继续使用二次探查，计算下一个散列地址：H2=（H0+2*2）%11=（2+4）%11=6，索引为6的位置存储的是关键字9，遇到冲突。

4.我们继续使用二次探查，计算下一个散列地址：H3=（H0+3*3）%11=（2+9）%11=0。索引为0的位置存储的是关键字11，遇到冲突。

5.我们继续使用二次探查，计算下一个散列地址：H4=（H0+4*4）%11=（2+16）%11=7，发现索引为7的位置是空的，确认查找失败，散列地址是7。

AcKing专业IT教育品牌
公众号 · AcKing

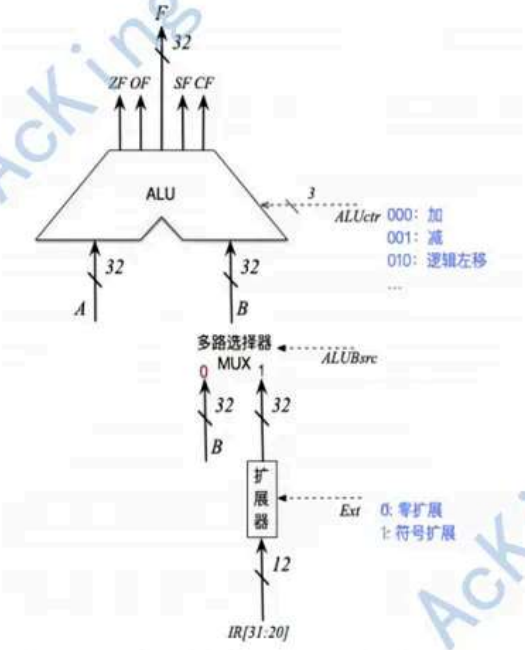
43（13分）假定计算机M字长为32位，按字节编址，采用32位定长指令字，指令add slli和lw的格式、编码和功能说明如图43（a）图所示。

二进制位序	31～25	24～20	19～15	14～12	11～7	6～0	指令功能说明
add 指令	0000000	rs2	rs1	000	rd	0110011	R[rd] ← R[rs1]+R[rs2]
slli 指令	0000000	shemt	rs1	010	rd	0010011	R[rd] ← R[rs1] << shemt
lw 指令	imm		rs1	010	rd	0000011	R[rd] ← M[R[rs1]+imm]

题43（a）图

其中R[x]表示通用寄存器x的内容，M[x]表示地址为x的存储单元内容，shamt为位移位数，imm为补码表示的偏移量。

题43（b）图给出了计算机M的部分数据通路及控制信号（用箭头虚线表示），其中，A和B分别表示从通用寄存器rs1和
rs2中读出的内容，IR[31:20]表示指令寄存器中的高12位；控制信号Ext为0、1时扩展器分别实现零扩展，符号扩展ALUctr为000、001、010时ALU分别加、减、逻辑左移运算。



(1)M最多有几个寄存器？为什么shemt占5位？

【答案】

从43题（a）图可以看到，寄存器rs1和rs2用5位表示，所以计算机M最多的是2^5=32个寄存器。

根据指令R[rd]<-R[rs1]<<shamt可知，shamt表示左移

shamt：计算机的机器字长是32位，所以shamt表示左移的最大范围不能超过32，所以只需要5bit就可以表示对应的范围：$\log_2 32=5$。

公众号 · AcKing

(2)执行add指令时，控制信号ALUBsrc的取值应该是什么？若rs1和rs2寄存器内容分别是87654321H和98765432H，则add指令指令后，ALU输出端F、OF和CF的结果分别是什么？若设add指令处理的是无符号整数，则应根据哪个标志判断是否溢出（5分）。

【答案】

ALUBsrc=0，它的作用是支持lw指令和imm的偏移（R[rd]<-M[R[rs1]+imm]）

F的计算如下：

```
  8765 4321
+ 9876 5432
-----------
 11FDB 9753
```

由于计算机的机器字长是32位，所以最高位的1舍掉，F=1FDB 9753H；

OF是溢出位，我们计算的结果发现确实发生了溢出和进位，所以OF=1。

CF是进位位，我们计算的结果发现确实发生了溢出和进位，所以CF=1。

溢出的判断方法是，最高位进位（异或）次高位，计算过程发现最高位进位，次高位没有进位，所以CF是标志判断是否溢出。

(3)执行slli指令时，控制信号Ext的取值可以是0也可以是1，为什么？（2分）

【答案】

slli代表左移指令，slli指令的高12位（即IR[31:20]）的最高位为0，因此无论进行零扩展还是符号扩展，都是在高位补0，效果等价，因此Ext可以是0也可以是1。

(4)执行lw指令时，控制信号Ext、ALUctr的取值分别是什么？（2分）

【答案】

R[rd]<-M[R[rs1]+imm]这条指令的意思是，rs1里的内容加上立即数，等于真正要访问的内存地址。

计算偏移地址，偏移地址可正可负，可以访问现在代码上边的代码，也可以访问下边的。

所以是符号扩展，Ext=1，然后表示加法，所以根据下图可知ALUctr=000

(5)若一条指令的机器码是A040 A103H，则该指令一定是lw指令，为什么？（2分）

【答案】

A040 A103H=1010000000100 00001 010 00010 0000011B，6~0位=0000011，中间的4~12位=010，最高的12位为A04H。其他两个指令add和slli的高12位都是000H，所以该指令一定是lw指令。

公众号 · AcKing

(6)若执行该指令时，R[01H]=FFFF A2D0H,则所读取数据的存储地址是多少？

【答案】

A040 A103H=101000000100 00001 010 00010 0000011B，6~0位=0000011，中间的4~12位=010，最高的12位为A04H。imm是31~25，也就是A404H（注意是符号扩展）

扩展完1111 1111 1111 1111 1111 1010 0000 0100，十六进制是FFFF FA04。rs1里的内容加上立即数，等于真正要访问的内存地址，R[01H]=FFFF A2D0H,所以FFFF FA04+FFFF A2D0=FFFF 9CD4H。

44. (10 分) 对于题 43 中的计算机 M，C 语言程序中包含的语句"`sum += a[i]`"，在 M 中对应的指令序列 S 如下：

```
slli r4, r2, 2     //R[r4] ← R[r2]<<2
add r4, r3, r4     //R[r4] ← R[r3]+R[r4]
lw r5, 0(r4)       //R[r5] ← M[R[r4]+0]
add r1, r1, r5     // R[r1] ← R[r1]+R[r5]
```

其中，i,sum, 数组 a 都为 int 型，r1 ~ r5 的寄存器编号为 01H ~ 05H 请回答下列问题：

(1) a 的首地址、变量 i、变量 sum 存储的寄存器编号是？

(2) 该系统采用小端方式，页式存储，页大小为 4KB。执行第 1 条指令时，i = 5，r1 = 0000 1332H，r3 = 0013 DFF0H，存储单元内容如下所示，执行 sum += a[i] 后，a[i] 的地址为？a[i] 和 sum 的机器数为？a[i] 所在页的页号为？此次执行中，数组 a 至少存放在几页中？

地址		0	1	2	3	4	5	6	7
0013	DFF0	FF	FF	FF	7C	70	FE	FF	FF
0013	DFF8	00	00	00	0C	3C	02	01	FF
0013	E000	F0	F1	00	00	DC	EC	FF	FF
0013	E008	FF	FF	01	02	00	00	01	02

(3) slli r4，r2，2 的机器码是？若 a 改为 short 型，则 slli 指令的汇编形式应该是？

(1)根据指令序列S中每条指令的功能，写出存放数组a的首地址、变量i和sum的通用寄存器编号（3分）

【答案】

slli r4,r2,2：这条指令将寄存器r2的值左移2位，结果存储在寄存器r4中。在C语言中，这对应于数组索引的计算（即i*4，因为每个int类型占4字节）。因此，我们可以推断出寄存器r2存储的是变量i的值，即i的寄存器编号位02H。

add r4,r3,r4：这条指令将寄存器r3和r4的值相加，结果存储在寄存器r4中。这对应于计算数组元素的内存地址（即&a[i]）。因此，我们可以推断出寄存器r3存储的是数组a的首地址，即a的寄存器编号为03H。

add r1,r1,r5：这条指令将寄存器r1和r5的值相加，结果存储在寄存器r1中。这对应于累加操作（即sum+=a[i]）。因此，我们可以推断出寄存器r1存储的是变量sum的值，即sum的寄存器编号为01H。

所以，数组a的首地址、变量i和sum的通用寄存器编号分别为03HH、02H和01H。

公众号 · AcKing

(2) 该系统采用小端方式，页式存储，页大小为 4KB。执行第 1 条指令时，i = 5，r1 = 0000 1332H，r3 = 0013 DFF0H，存储单元内容如下所示，执行 sum += a[i] 后，a[i] 的地址为？a[i] 和 sum 的机器数为？a[i] 所在页的页号为？此次执行中，数组 a 至少存放在几页中？

地址		0	1	2	3	4	5	6	7
0013	DFF0	FF	FF	FF	7C	70	FE	FF	FF
0013	DFF8	00	00	00	0C	3C	02	01	FF
0013	E000	F0	F1	00	00	DC	EC	FF	FF
0013	E008	FF	FF	01	02	00	00	01	02

【答案】

执行“sum+=a[i]”语句后，i=6,我们直到i占32位所以一次占4个位置。i=0时，占FF FF FF 7C

i=1时，占70 FE FF FF

i=2时，占00 00 00 0C

i=3时，占3C 02 01 FF

i=4时，占FF FF FF 7C

i=5时，占F0 F1 00 00

i=6时，占DC EC FF FF

所以a[i]的地址=0013 E000H+第四个地址=0013 E004H

a[i]的机器数按照小端编制，所以DC作为最低位放在最右边，以此类推可得：a[i]的机器数=FFFF FCDCH。

sum的机器数=0000 1332H + FFFF ECDCH= 1 0000 000EH。由于只有23位，所以最高位舍掉后答案为0000 000EH。

页大小为4KB=2^{12}B，所以页内地址占12位，去掉后12位剩余的则是20位页号，a[i]所在页页号=0013EH。

我们有20位页号，根据题目可知数组跨页号了0013e和0013d，所以数组a至少存放在2页中。

(3)指令“slli r4,r2,2”的机器码是什么（用十六进制表示）？若数组a改为short类型，则指令序列存在S中slli指令的汇编形式应是什么？

【答案】

指令 31 25 24 20 19 15 14 12 11 6 0

slli 000000 shamt rs1 010 rd 0010011 R[rd]<-R[rs1]<<shamt

slli r4,r2,2 //R[r4]<-R[r2]<<2

通用寄存器r1-r5的编号位01H-05H。

6~0：由表可得：0010011

11~7：rd=r2=00010

14~12：由表可得：010

19~15：rd=r4=00100

shamt=2，所以24~20：0010

公众号 · AcKing

31~25由表可得：00000000
机器码=0000 0000 0010 0001 0010 00100 001 0011=0021 2313H。
若a改为short类型，slli指令的汇编形式应该是slli r4,r2,1。

45. (7 分) 某计算机采用虚拟页式存储管理，虚拟地址、物理地址为 32 位，页表项大小为 4B，页面大小为 4MB。虚拟地址结构如下：

页号（10 位）	页内偏移量（22 位）

进程 P 的页表起始虚拟地址为 B8C0 0000H，页表被装载到物理地址为 6540 0000H 开始的连续空间。请回答下列问题：

(1) 进程 P 访问虚拟地址 1234 5678H 时发生缺页，经缺页异常处理和 MMU 地址转换之后，得到的物理地址 =BAB4 5678H，此次缺页异常处理过程中，需要为所缺页面分配页框，并更新相应的页表项，则该页表项的虚拟地址、物理地址分别是？该页表项的页框号更新后的值是什么？

【答案】
首先，我们需要确定虚拟地址1234 5678H对应页号。由于页号占10位，1234 5678H=0001 0010 0011 0100 0101 0110 0111 1000。
计算得到：
页内偏移量（22位）=11 0100 0101 0110 0111 1000=35678H
页号（10位）=00 0100 1000=048H
然后，我们需要找到这个页号对应的页表项的虚拟地址和物理地址。由于页表项的大小为4字节，我们可以通过将页号乘以4得到页表项的偏移量。然后将这个偏移量加到页表的起始地址上，就可以得到页表项的虚拟地址和物理地址。进程P的页表起始虚拟地址为B8C0 0000H，物理地址6540 0000H
计算得到：页表项虚拟地址=页表起始虚拟地址+页号*4=B8C00000H+048H*4=B8C0 0120H，页表项物理地址=页表起始物理地址+页号*4=65400000H+048H*4=6540 0120H。
最后，我们需要更新页表项中的页框号。由于经过MMU地址转换后得到的物理地址是BAB4 5678H，我们可以通过右移22位得到页框号。
计算得到：页框号=物理地址BAB4 5678H的前10位，即1011101010 B=2EAH。

(2) 进程 P 的页表所在页的页号是多少？该页对应的页表项的虚拟地址是多少？该页表项中的页框号是多少？

【答案】
首先，我们需要确定进程P的页表所在页的页号。由于页表起始虚拟地址位B8C0 0000H，我们可以通过右移22位得到页号。
计算得到：
进程P的页表所在页的页号等于B8C0 0000H的前10位，即1011100011B=2E3H。
然后，我们需要找到这个页号对应的页表项的虚拟地址。由于页表项的大小为4字节，我们可以通过将页号乘以4得到页表项的偏移量。然后将这个偏移量加到页表的起始地址上，就可以得到页表项的虚拟地址。
计算得到：
该页对应的页表项的虚拟地址=B8C0 0000H + 2E3H*4=B8C0 0B8CH。
最后，我们需要确定页表项中的页框号。由于页表被装在到从物理地址6540 0000H开始的连续主存空间中，我们可以通过右移22位得到页框号。
计算得到：
该页表项中的页框号等于物理地址6540 0000H的前10位，即0110 0101 01B=195H。

公众号 · AcKing

46（8分）计算机系统中的进程之间往往需要相互协作以完成一个任务，在某网络系统中缓冲区B用于存放一个数据分组，对B的操作有C1、C2和C3。C1将一个数据分组写入B中，C2从B中读出一个数据分组，C3对B中的数据分组进行修改。要求B为空时才能执行C1，B非空时才能执行C2和C3。请回答下列问题。

（1）假设进程P1和P2均需执行C1，实现C1的代码是否为临界区？为什么？

【答案】

是的，实现C1的代码可以被视为临界区。临界区是指在并发编程中，当多个进程同时访问和修改共享数据时，必须进行互斥访问的代码区域。

在这个例子中，进程P1和P2都需要执行C1，即它们都需要将一个数据分组写入缓冲区B。如果这两个进程同时执行C1，那么它们可能会试图同时写入数据分组，这可能会导致数据的不一致性。因此，我们需要确保在任何时刻，只有一个进程可以执行C1。这就需要将执行C1的代码区域定义为临界区，并使用适当的同步机制（如互斥锁或信号量）来保证在同一时刻只有一个进程可以进入临界区。

所以，实现C1的代码是临界区，因为它涉及到对共享资源（在这里是缓冲区B）的修改，而这个修改需要被同步，以防止数据的不一致性。

（2）假设B初始为空，进程P1执行C1一次，进程P2执行C2一次。请定义尽可能少的信号量，并用wait（），signal（）操作描述进程P1、P2之间的同步或互斥关系，说明所用信号量的作用及初值。（3分）

【答案】

在这个问题中，我们可以使用两个信号量：一个用于保护缓冲区B（我们称之为mutex），另一个用于同步进程P1和P2（我们称之为full。）mutex用于确保在同一时刻只有一个进程可以访问缓冲区B，而full用于表示缓冲区B是否已满。

初始时，mutex的值为1，表示缓冲区B是可用的；full的值为0，表示缓冲区B是空的。

以下是进程P1和P2的代码：

```
//进程P1
P1(){
    wait(mutex);  //请求访问缓冲区B
    //执行C1，将一个数据分组写入B中
    signal(mutex);//释放对缓冲区B的访问
    signal(full); //表示缓冲区B已满
}
```

公众号 · AcKing

```
//进程P2
P2(){
   wait(full);//等待缓冲区B变为满
   wait(mutex);//请求访问缓冲区B
   //执行C2，从B中读出一个数据分组
   signal(mutex);//释放对缓冲区B的访问
}
```

在这个代码中，wait()操作表示请求一个信号量，如果信号量的值大于0，那么就将其减1；如果信号量的值为0，那么就阻塞，直到信号量的值大于0。signal()操作表示释放一个信号量，将其值加1。

（3）假设B初始不为空，进程P1和P2各执行C3一次，请定义尽可能少的信号量，并用wait()、signal()操作描述进程P1和P2之间的同步或互斥关系，说明所用信号量的作用及初值。（3分）

【答案】

在这个问题中，我们可以使用一个信号量：一个用于保护缓冲区B（我们称之为mutex）。mutex用于确保在同一时刻只有一个进程可以访问缓冲区B。

初始时，mutex的值为1，表示缓冲区B是可用的。

以下是进程P1和P2的代码：

```
//进程P1
P1(){
    wait(mutex);    //请求访问缓冲区B
    //执行C3，对B中的数据分组进行修改
    signal(mutex); //释放对缓冲区B的访问
}
//进程P2
P2(){
    wait(mutex);  //请求访问缓冲区B
    //执行C3，对B中的数据分组进行修改
    signal(mutex);//释放对缓冲区B的访问
}
```

在这个代码中，wait()操作表示请求一个信号量，如果信号量的值大于0，那么就将其减1；如果信号量的值为0，那么就阻塞，直到信号量的值大于0。signal()操作表示释放一个信号量，将其值加1。

所以，实现C3的代码是临界区，因为它涉及到共享资源（在这里是缓冲区B）的修改，而这个修改需要被同步，以防止数据的不一致性。

公众号 · AcKing

47. 给了 4 个自治系统，一个自治系统路由器少于 15 个，一个多于 20 个，最下方的自治系统有一个具体的路由结构表。请回答下列问题：

(1) 当自治系统内路由器个数大于 20 时，自治系统内路由选择算法应当采用 OSPF 还是 RIP?

(2) 起始交换并建立路由表，每次交换耗时 30s。不考虑初始建立交换，从第一次正式交换起，经过多长时间下图中所有路由器都可以收到左下网络的到达路径信息？多久可以全部接收到右下网络到达路径信息？

(3) 右上方的自治系统检测到了一个直连网络，将这个信息通报给左下方的一个自治系统；第 1 问是右上方发给右下方的 BGP 会话是什么？第 2 问是用的是什么 BGP 报文？第 3 问是自治系统边界到自治系统内通知信息用的是 BGP 哪一个会话？

(4) 给了三个 BGP 通报路径，一个长度 3，一个长度 5，一个长度 3，大概就是 AS1、AS2、AS3 这样的路径，然后问的是最下方的自治系统中，偏左边和偏右边的路由器 R12、R14 如果要通过 BGP 路径到达最上方一个网络，下一跳应该是哪一个路由器？

本题无图，暂无解析

AcKing专业IT教育品牌
公众号 · AcKing