2008年硕士学位研究生入学考试试题

（计算机系统结构）

所有试题答案写在答题纸上，答案写在试卷上无效

第一部分 数据结构，共70分

一、是非题（每小题1.5分，共12分）

1. 任一字符的编码不是另一字符的编码的前缀，则这种编码称做前缀编码。

2. Dijkstra算法是一个按路径长度递增的次序产生最短路径的算法。

3. 若一棵二叉树中所有结点的平衡因子不大于1，则该二叉树是平衡二叉树。

4. 在使用边界标识法进行动态存储管理时，分配可按首次拟合进行，也可按最佳拟合进行。

5. 对于顺序文件，在存取第i个记录时，不必先搜索在它之前的i-1个记录。

6. 要想在O(1)的时间内访问单链表的第一个结点或最后一个结点，这个单链表必须是用尾指针表示的单循环链表。

7. 任何一个中缀算术表达式都可以转换成一个不含括号的波兰式。

8. 顺序表、循环队列、哈希表都是与数据的存储结构有关的术语。

二、问答题（每小题5分，共20分）

1.（5分）假设对具有7个结点的二叉树中的各结点从1到7加以编号，若其先序遍历序列为1,2,...,7，问其中序遍历序列可能有多少种？

2.（5分）试写出表达式(a+b)*c+(d-c)*(a+b)/a的波兰式。

3.（5分）对n个元素进行快速排序，其最大递归深度是多少？最小递归深度是多少？试分别说明两种情况下待排序的元素具有什么特征。

4.（5分）若完全二叉树有34个叶结点，则该二叉树最多有多少个结点？处于二叉树中最下面一层的叶子结点数是多少？请说明推导理由。

三、计算题（共 22 分）

1.（12 分）已知一个无向网包含有 a, b, c, d, e, f, g, h 八个顶点，其邻接矩阵如右图所示，要求：

1) 请画出该网；

2) 画出基于该邻接矩阵的网 G 的宽度优先搜索生成树；

3) 按克鲁斯卡尔算法给出 G 的一棵最小生成树的生成过程(要求给出步骤)。

	a	b	c	d	e	f	g	h
a	∞	4	3	∞	∞	∞	∞	∞
b	4	∞	5	5	9	∞	∞	∞
c	3	5	∞	5	∞	∞	∞	5
d	∞	5	5	∞	7	6	5	4
e	∞	9	∞	7	∞	3	∞	∞
f	∞	∞	∞	6	3	∞	2	∞
g	∞	∞	∞	5	∞	2	∞	6
h	∞	∞	5	4	∞	∞	6	∞

2. (10 分)将下列树的孩子-兄弟链表改为后根遍历线索化链表。

下标	1	2	3	4	5	6	7	8	9	10	11
data	A	B	C	D	E	F	G	H	I	J	K
ltag	0	0	0	0	0	0	0	0	0	0	0
firstchild	2	4	5	0	0	8	0	0	0	0	0
rtag	0	0	0	0	0	0	0	0	0	0	0
nextsbling	0	3	7	0	6	9	11	0	10	0	0

四、算法设计题（每小题 8 分，共 16 分）

1. 已知二叉树以二叉链表为存储结构，其结点结构定义如下：

```
typedef struct BiTNode{
    ElemType        data;
    struct BiTNode   *lchild, *rchild;
    int              leftsize;
}BiTNode, *BiTree;
```

其中 leftsize 域存储该结点的左子树中的结点个数。开始时，leftsize 域值均为 0，T 为指向某二叉树根结点的指针。请编写一算法填写该二叉树中每个结点的 leftsize 域。

2．编写一算法，判断以邻接表方式存储的有向图中是否存在顶点 v_i 到顶点 v_j（$i \neq j$）的路径。

第二部分 操作系统，共 40 分

一、问答题（共 25 分）

1. （5分）什么是系统调用？在UNIX中用于打开文件的open系统调用绝对需要吗？如果没有会产生什么后果？

2. （5分）I/O软件通常组织成哪四个层次？每一层的主要功能是什么？

3. （5分）在硬件中断完成后，操作系统的中断处理程序主要完成哪些工作？

4. （10分）虚拟存储系统中的页面置换算法是利用局部性原理来预测进程地址空间中哪些页面是未来不再使用的或短期内较少使用的。试问：什么是局部性原理？试说明时钟(Clock)页面置换算法和工作集时钟（WSClock）页面置换算法的工作过程。

二、设计题（共15分）

公共汽车司机与售票员之间必须协同工作。他们的协作要遵守下面的操作规则：
A.只有售票员把所有车门关好了，司机才能开车；
B.只有当汽车已完全停稳后，售票员才能开门上下乘客；
C.通过询问和观察，售票员能在开门前知道是否有乘客上下。在没有乘客上下车时，售票员可以不开门，直接通知司机开车。
（1）假定某辆公共汽车上有一名司机、一名售票员和一个车门。试用C语言实现司机进程和售票员进程，以遵守上述操作规则；
（2）假定某辆公共汽车上有一名司机、两名售票员和两个车门。试用C语言实现司机进程和售票员进程，以遵守上述操作规则。
要求：用信号量方法（不允许使用信号量数组），并给出信号量的定义和初始值；在代码中要有适当的注释。

第三部分 计算机组成原理，共40分

一、简答题（每小题4分，共24分）

1. 机器指令字由哪几部分组成？指令集中通常包括哪些操作？典型的寻址方式有哪些？

2. 图示说明时钟周期、总线周期、机器周期和指令周期之间的关系；

3. 说明Cache的容量和块大小与其命中率的关系；

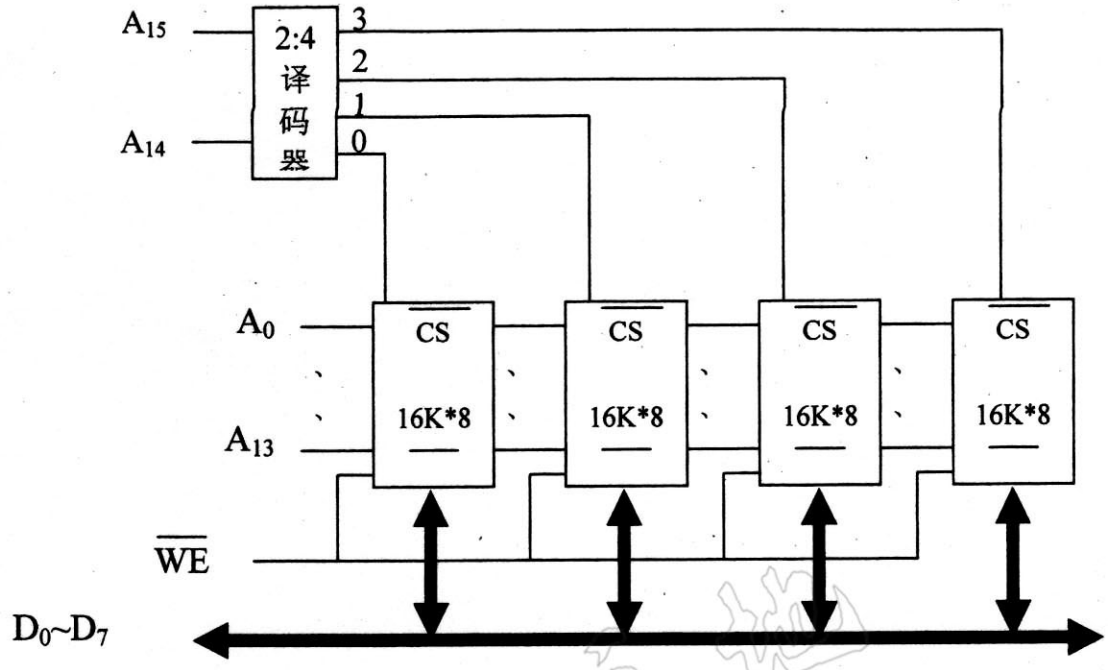
4. 列举机器数的表示形式及其特点；

5. 中断接口由哪些部分组成？说明处理器响应中断的过程；

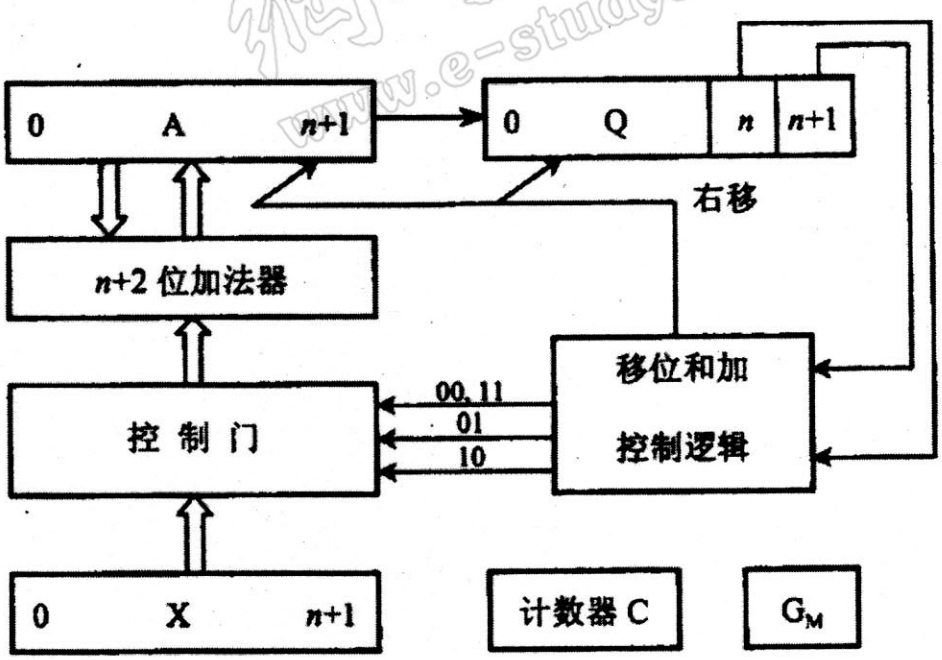
6. 说明以组合逻辑方式实现处理器控制器的设计步骤。

二、综合题（每题8分，共16分）

1. 某存储系统结构如下图所示，计算各个存储体的地址范围。



2. 下图为补码比较法（Booth 算法）所需的硬件配置，据此给出完成一位乘法的计算步骤。



2008年硕士学位研究生入学考试试题参考答案

（计算机系统结构）

所有试题答案写在答题纸上，答案写在试卷上无效

第一部分 数据结构，共70分

一、是非题（每小题1.5分，共12分）

1.对 2.对 3.错 4.对 5.错 6.错 7.对 8.错

二、问答题（每小题5分，共20分）

1. $b_7 = \frac{1}{8}\frac{14!}{7!7!} = 429$

2. +*+abc/*-dc+aba

3.最大递归深度为 n（如待排序的元素按正序排列），最小递归深度为 $\lfloor \log_2 n \rfloor + 1$（每一趟的枢轴经划分后都调整到当前待排序元素序列的中间）

4.最多有68个结点，最下面一层的叶子结点数为5。完全二叉树最多有一个度为1的结点，度为2的结点数比叶子结点数少1。

三、计算题（共22分）

1. （12分）略

2. （10分）

下标	1	2	3	4	5	6	7	8	9	10	11
data	A	B	C	D	E	F	G	H	I	J	K
ltag	0	0	0	1	1	0	1	1	1	1	1
firstchild	2	4	5	0	2	8	3	5	6	9	7
rtag	1	0	0	1	0	0	0	1	0	1	1
nextsbling	0	3	7	2	6	9	11	6	10	3	1

四、算法设计题（每小题8分，共16分）

1．可以基于中序遍历，在访问结点时需要统计以该结点为根的二叉树中的结点数目。

2．从 v_i 开始进行深度优先遍历看是否能访问到 v_j。

第二部分 操作系统，共40分

一、 问答题（共25分）

1.（5分）什么是系统调用？在UNIX中用于打开文件的open系统调用绝对需要吗？如果没有会产生什么后果？

答：系统调用是用户程序请求操作系统服务的一组接口程序。在使用文件系统之前，必须先打开文件，指定要打开的文件名以及打开方式，并把文件属性和硬盘地址表装入主存，便于后续的读写系统调用快速地存取。否则，后续的读写操作就无法高效地实现。

2.（5分）I/O软件通常组织成四个层次？每一层的主要功能是什么？

答:I/O软件通常组织成四个层次。从底层向上层依次是：（1）中断处理程序。主要功能是当I/O完成时唤醒相应的设备驱动程序；（2）设备驱动程序。主要功能是接收来自上层与设备无关的操作系统程序发出的抽象的读写请求，检查设备状态，设置设备寄存器；（3）与设备无关的操作系统程序。主要功能是执行对所有设备公共的I/O功能，包括命名、保护、分块、缓冲、分配和释放设备，并且向用户层I/O软件提出供一个统一的接口；（4）用户层I/O软件。主要功能是产生I/O请求，对I/O进行格式化处理，实现假脱机等。

3.（5分）在硬件中断完成后，操作系统的中断处理程序主要完成哪些工作？

答:中断处理程序主要完成如下一些工作：（1）切换上下文。即保存所有被中断进程的通用寄存器和状态寄存器的值，设置中断程序执行时需要的所有寄存器值和栈；（2）运行中断服务程序；（3）应答中断控制器；（4）选择下一个要运行的进程，并为它设置所需要的全部上下文；（5）运行下一个被选中的进程。

4.（10分）虚拟存储系统中的页面置换算法是利用局部性原理来预测进程地址空间中哪些页面是未来不再使用的或短期内较少使用的。试问：什么是局部性原理？试说明时钟(Clock)页面置换算法和工作集时钟（WSClock）页面置换算法的工作过程。

答:

局部性原理

局部性原理有两个方面的含义：（1）进程在一段时间里会顺序执行邻近的代码或反复执行相同的代码，称之为程序局部性；（2）进程在一段时间里会顺序访问邻近的数据或反复访问相同的数据集，称之为数据局部性。一段时间里反复执行相同的代码或反复访问相同的数据集又称为时间局部性，顺序访问邻近的代码或顺序访问邻近的数据又称为空间局部性。

Clock页面置换算法的工作过程

系统为每个页面设置R位，用以记录其是否被访问过。当启动一个进程时，系统将所有页面的R位清0，并定期地将R位清0。每当某个页被访问一次，则将其R位置为1。Clock页面置换算法的工作过程是：把所有的页面保存在一个类似钟面的环形链表中，有一个表指针指向最老的页面。当发生页面失效时，算法首先检查表针指向的页面，如果它的R位是0就淘汰该页面，并把新的页面插入这个位置，然后把表针前移一个位置；如果R位是1则将R位清0，并把表针前移一个位置。重复这个过程直到找到一个R位为0的页面为止。

工作集时钟页面置换算法的工作过程

与Clock算法一样，所需的数据结构是一个以页为元素的循环表。最初，该表是空的。当装入第一个页面后，把它加入到该表中。随着更多的页面加入该表，它们形成一个环。每个表项包含来自基本工作集算法的上次使用时间域，以及已标记的R位。与Clock算法一样，

每次页面失效时，首先检查指针指向的页面。如果R位已被置为1，说明该页面在当前时钟滴答中被使用过，那么该页面就不适合被淘汰。然后把该页面的R位清0，指针指向下一个页面，重复这个过程直到找到一个R位为0的页面为止。

二、设计题（共15分）

公共汽车司机与售票员之间必须协同工作。他们的协作要遵守下面的操作规则：

A. 只有售票员把所有车门关好了，司机才能开车；

B. 只有当汽车已完全停稳后，售票员才能开门上下乘客；

C. 通过询问和观察，售票员能在开门前知道是否有乘客上下。在没有乘客上下车时，售票员可以不开门，直接通知司机开车。

（1）假定某辆公共汽车上有一名司机、一名售票员和一个车门。试用C语言实现司机进程和售票员进程，以遵守上述操作规则；

（2）假定某辆公共汽车上有一名司机、两名售票员和两个车门。试用C语言实现司机进程和售票员进程，以遵守上述操作规则。

要求：用信号量方法（不允许使用信号量数组），并给出信号量的定义和初始值；在代码中要有适当的注释。

答：设 driver 为司机进程，conductor 为售票员进程；bus-stoped 和 door-closed 分别为司机和售票员的私有信号量，mutex 为控制互斥的信号量。

（1）则司机和售票员的同步过程可描述如下：

```
typedef int semaphore;
semaphore door-closed=0;     /*售票员的信号量*/
semaphore bus-stoped=0;      /*司机的信号量*/
int PassengerNumber=0;       /*等待上下的乘客数*/
void driver(void){
        while(TRUE){
            driving_bus();
            stop_bus();
            up (bus-stoped) ;
            down (door-closed) ;
            driving_bus();
        }
}
void conductor(void){
        while(TRUE){
            ticketing();
            PassengerNumber=get_passenger_num(); /*售票员获知是否有乘客上下*/
            if (PassengerNumber <>0){    /*有乘客上下*/
                down (bus-stoped) ;
                open_door();             /*开门*/
                waiting_passenger();     /*等待乘客上下完毕*/
                close_door();            /*关门*/
                up (door-closed) ;
                }
            else{ /*没有乘客上下*/
                down (bus-stoped) ;
```

```
                up (door-closed) ;
                }
        }
}
```

（2）则司机和售票员的同步过程描述如下：

```
typedef int semaphore;
semaphore door-closed=0;      /*售票员的信号量*/
semaphore bus-stoped=0;       /*司机的信号量*/
semaphore mutex=1;            /*控制互斥的信号量*/
int PassengerNumber=0;        /*等待上下的乘客数*/
int opened_door=0;            /*被打开的门数*/

void driver(void){
        while(TRUE){
            driving_bus();
            stop_bus();
            up (bus-stoped) ;
            down (door-closed) ;
            driving_bus();
        }
}
void conductor(int i){/*i 为售票员的编号*/
        while(TRUE){
            ticketing();
            down (mutex) ;
            PassengerNumber=get_passenger_num() ;        /*售票员获知是否有乘客
        上下*/
            if (PassengerNumber() <>0){                  /*有乘客上下*/
                down (bus-stoped) ;
                open_door();                             /*开门*/
                opened_door = opened_door+1;
                if (opened_door==1) up (bus-stoped) ;    /*如果这是第一个被打开
            的门*/
                up (mutex) ;
                waiting_passenger();                     /*等待乘客上下完毕*/
                close_door();                            /*关门*/
                down (mutex) ;
                opened_door = opened_door-1;
                if (opened_door==0) up (door-closed) ; /*如果所有的门被关闭*/
                up (mutex) ;
                }
            else{
                down (bus-stoped) ;
                up (door-closed) ;
```

```
                up（mutex）；
                }
        }
}
```

第三部分 计算机组成原理，共 40 分

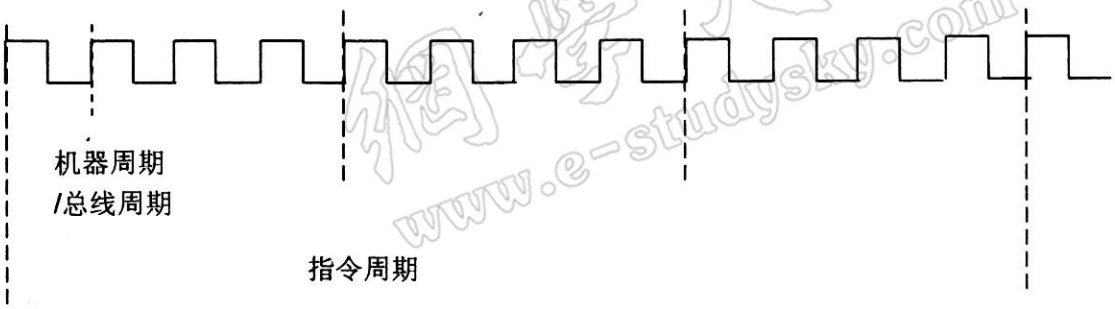
一、简答题（每小题 4 分，共 24 分）

1．机器指令字由哪几部分组成？指令集中通常包括哪些操作？典型的寻址方式有哪些？

答：操作码，地址码；数据传输，算术逻辑运算，I/O，转移控制，系统操作；10 种。

2．解释时钟周期、总线周期、机器周期、指令周期，并图示说明它们之间的关系；

答：



3．说明 Cache 的容量和块大小与其命中率的关系；

答：容量增加，命中率增加，到一定容量后，命中率不变；块大小增加，命中率增加，到一定大小后，命中率下降。

4．列举机器数的表示形式及其特点；

答：定点、浮点、原码、补码、反码、移码

5．中断接口由哪些部分组成？说明处理器响应中断的过程；

答：地址寄存器、数据寄存器、ISR、IMR 等

中断向量，保存断点，保存现场，中断处理，恢复现场，中断返回

6．说明以组合逻辑方式实现处理器控制器的设计步骤。

答：确定指令集，确定体系结构，确定时序，确定各个机器周期的指令微操作，确定各个节拍微操作分配，构建逻辑表达式，生成控制逻辑

二、综合题（每题 8 分，共 16 分）

1．某存储系统结构如下图所示，计算各个存储体的地址范围。

地址片号	A15A14	A13A12A11…A1A0	说明
1	00 00	000…00 111…11	最低地址 0000H 最高地址 3FFFH
2	01 01	000…00 111…11	最低地址 40000H 最高地址 7FFFH
3	10 10	000..00 111…11	最低地址 8000H 最高地址 0BFFFH
4	11 11	000…00 111…11	最低地址 0C000H 最高地址 0FFFFH

2. 下图为补码比较法（Booth 算法）所需的硬件配置，据此给出完成一位乘法的计算步骤。