

计算机学科专业基础综合试题

参考答案(2017 年)

一、单项选择题

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

二、综合应用题

41. 【答案要点】

(1) 算法的基本设计思想

表达式树的中序序列加上必要的括号即为等价的中缀表达式。可以基于二叉树的中序遍历策略得到所需的表达式。(3分)

表达式树中分支结点所对应的子表达式的计算次序,由该分支结点所处的位置决定。为得到正确的中缀表达式,需要在生成遍历序列的同时,在适当位置增加必要的括号。显然,表达式的最外层(对应根结点)及操作数(对应叶结点)不需要添加括号。(2分)

(2) 算法实现(10分)

```

void BtreeToE(BTree * root)
{
    BtreeToExp(root, 1);          //根的高度为 1
}

void BtreeToExp(BTree * root, int deep)
{
    if( root == NULL) return;
    else if( root->left == NULL && root->right == NULL)
        // 若为叶结点
        printf( "%s" , root ->data);    // 输出操作数
    else
    {
        if( deep>1)  printf( "(" );// 若有子表达式则加 1 层括号
        BtreeToExp( root->left, deep+1);
        printf( "%s" , root ->data);    // 输出操作符
        BtreeToExp( root->right, deep+1);
        if( deep>1)  printf( ")" );//若有子表达式则加 1 层括号
    }
}

```

【评分说明】

- ① 若考生设计的算法满足题目的功能要求,则(1)、(2)根据所实现算法的策略及输出结果给分,细则见下表。

| 分数 | 备注 |
|----|---|
| 15 | 采用中序遍历算法且正确,括号嵌套正确,层数适当。 |
| 14 | 采用中序遍历算法且正确,括号嵌套正确,但括号嵌套层数过多。例如,表达式最外层加上括号,或操作数加括号如(a)。 |
| 11 | 采用中序遍历算法,但括号嵌套层数不完全正确。例如,左右括号数量不匹配。 |

续表

| 分数 | 备注 |
|----------|-------------------|
| 9 | 采用中序遍历算法,但没有考虑括号。 |
| ≤ 7 | 其他 |

- ② 若考生采用其他方法得到正确结果,可参照①的评分标准给分。
- ③ 如果程序中使用了求结点深度等辅助函数,但没有给出相应的实现过程,只要考生进行了必要的说明,可不扣分。
- ④ 若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路,但在算法实现中能够表达出算法思想且正确的,可参照①的标准给分。
- ⑤ 若算法的基本设计思想描述或算法实现中部分正确,可参照①中各种情况的相应给分标准酌情给分。
- ⑥ 参考答案中只给出了使用 C 语言的版本,使用 C++语言的答案参照以上评分标准。

42. 【答案要点】

(1) 依次选出的边为:

(A,D),(D,E),(C,E),(B,C) (4分)

【评分说明】每正确选对一条边且次序正确,给1分。若考生选择的边正确,但次序不完全正确,酌情给分。

(2) 图 G 的 MST 是唯一的。(2分)

(3) 当带权连通图的任意一个环中所包含的边的权值均不相同,其 MST 是唯一的。(2分)

【评分说明】

- ① 若考生答案中给出的是其他充分条件,例如“带权连通图的所有边的权值均不相同”,同样给分。
- ② 若考生给出的充分条件对图的顶点数和边数做了某些限制,例如,限制了图中顶点的个数(顶点个数少于3个)、限制了图的形状(图中没有环)等,则最高给1分。

③ 答案部分正确,酌情给分。

43. 【答案要点】

(1) 由于 i 和 n 是 unsigned 型,故“ $i \leq n-1$ ”是无符号数比较, $n=0$ 时, $n-1$ 的机器数为全 1,值是 $2^{32}-1$,为 unsigned 型可表示的最大数,条件“ $i \leq n-1$ ”永真,因此出现死循环。(2 分)

若 i 和 n 改为 int 类型,则不会出现死循环。(1 分)

因为“ $i \leq n-1$ ”是带符号整数比较, $n=0$ 时, $n-1$ 的值是 -1,当 $i=0$ 时条件“ $i \leq n-1$ ”不成立,此时退出 for 循环。(1 分)

(2) $f1(23)$ 与 $f2(23)$ 的返回值相等。(1 分)

$f1(23)$ 的机器数是 00FF FFFFH,(1 分)

$f2(23)$ 的机器数是 4B7F FFFFH。(1 分)

(3) 当 $n=24$ 时, $f(24)=1\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ B$,而 float 型数只有 24 位有效位,舍入后数值增大,所以 $f2(24)$ 比 $f1(24)$ 大 1。(1 分)

【评分说明】只要说明 $f2(24)$ 需舍入处理即可给分。

(4) 显然 $f(31)$ 已超出了 int 型数据的表示范围,用 $f1(31)$ 实现时得到的机器数为 32 个 1,作为 int 型数解释时其值为 -1,即 $f1(31)$ 的返回值为 -1。(1 分)

因为 int 型最大可表示数是 0 后面加 31 个 1,故使 $f1(n)$ 的返回值与 $f(n)$ 相等的最大 n 值是 30。(1 分)

【评分说明】对于第二问,只要给出 $n=30$ 即可给分。

(5) IEEE 754 标准用“阶码全 1、尾数全 0”表示无穷大。 $f2$ 返回值为 float 型,机器数 7F80 0000H 对应的值是 $+\infty$ 。(1 分)

当 $n=126$ 时, $f(126)=2^{127}-1=1.1\cdots 1\times 2^{126}$,对应阶码为 $127+126=253$,尾数部分舍入后阶码加 1,最终阶码为 254,是 IEEE 754 单精度格式表示的最大阶码。故使 $f2$ 结果不溢出的最大 n 值为 126。(1 分)

当 $n=23$ 时, $f(23)$ 为 24 位 1, float 型数有 24 位有效位,所以不需舍入,结果精确。故使 $f2$ 获得精确结果的最大 n 值为 23。

(1 分)

【评分说明】对于第二问,只要给出 $n=23$,即可给分。对于第三问,只要给出 $n=126$,即可给分。

44. 【答案要点】

(1) M 为 CISC。(1 分)

M 的指令长短不一,不符合 RISC 指令系统特点。(1 分)

(2) f1 的机器代码占 96 B。(1 分)

因为 f1 的第一条指令“push ebp”所在的虚拟地址为 0040 1020H,最后一条指令“ret”所在的虚拟地址为 0040 107FH,所以,f1 的机器指令代码长度为 $0040\ 107FH - 0040\ 1020H + 1 = 60H = 96$ 个字节。(1 分)

(3) CF=1。(1 分)

cmp 指令实现 i 与 $n-1$ 的比较功能,进行的是减法运算。在执行 f1(0)过程中, $n=0$,当 $i=0$ 时, $i=0000\ 0000H$,并且 $n-1=FFFF\ FFFFH$ 。因此,当执行第 20 条指令时,在补码加/减运算器中执行“0 减 FFFF FFFFH”的操作,即 $0000\ 0000H + 0000\ 0000H + 1 = 0000\ 0001H$,此时,进位输出 $C=0$,减法运算时的借位标志 $CF=C \oplus 1=1$ 。(2 分)

(4) f2 中不能用 shl 指令实现 $power * 2$ 。(1 分)

因为 shl 指令用来将一个整数的所有有效数位作为一个整体左移;而 f2 中的变量 power 是 float 型,其机器数中不包含最高有效数位,但包含了阶码部分,将其作为一个整体左移时并不能实现“乘 2”的功能,因而 f2 中不能用 shl 指令实现 $power * 2$ 。(2 分)

45. 【答案要点】

(1) 函数 f1 的代码段中所有指令的虚拟地址的高 20 位相同,因此 f1 的机器指令代码在同一页中,仅占用 1 页。(1 分)

(2) push ebp 指令的虚拟地址的最高 10 位(页目录号)为 00 0000 0001,中间 10 位(页表索引)为 00 0000 0001,所以,取该指令时访问了页目录的第 1 个表项,(1 分)在对应的页表中访问了

第1个表项。(1分)

- (3) 在执行 `scanf()` 的过程中,进程 P 因等待输入而从执行态变为阻塞态。(1分)输入结束时,P 被中断处理程序唤醒,变为就绪态。(1分)P 被调度程序调度,变为运行态。(1分)CPU 状态会从用户态变为内核态。(1分)

46. 【答案要点】

`semaphore mutex_y1 = 1;` // `mutex_y1` 用于 `thread1` 与 `thread3` 对变量 `y` 的互斥访问。(1分)

`semaphore mutex_y2 = 1;` // `mutex_y2` 用于 `thread2` 与 `thread3` 对变量 `y` 的互斥访问。(1分)

`semaphore mutex_z = 1;` // `mutex_z` 用于变量 `z` 的互斥访问。(1分)
互斥代码如下:(5分)

| thread1 | thread2 | thread3 |
|---|---|--|
| <pre> { cnum w; wait(mutex_y1); w = add(x, y); signal(mutex_y1); } </pre> | <pre> { cnum w; wait(mutex_y2); wait(mutex_z); w = add(y, z); signal(mutex_z); signal(mutex_y2); } </pre> | <pre> { cnum w; w.a = 1; w.b = 1; wait(mutex_z); z = add(z, w); signal(mutex_z); wait(mutex_y1); wait(mutex_y2); y = add(y, w); signal(mutex_y1); signal(mutex_y2); } </pre> |

【评分说明】

- ① 各线程与变量之间的互斥、并发情况及相应评分见下表。

| 线程对 变量 | thread1 和 thread2 | thread2 和 thread3 | thread1 和 thread3 | 给分 |
|-----------|----------------------|----------------------|----------------------|-----|
| x | 不共享 | 不共享 | 不共享 | 1 分 |
| y | 同时读 | 读写互斥 | 读写互斥 | 3 分 |
| z | 不共享 | 读写互斥 | 不共享 | 1 分 |

② 若考生仅使用一个互斥信号量,互斥代码部分的得分最多给 2 分。

③ 答案部分正确,酌情给分。

47. 【答案要点】

- (1) t_0 时刻到 t_1 时刻期间,甲方可以断定乙方已正确接收了 3 个数据帧,(1 分)分别是 S0,0、S1,0、S2,0。(1 分)
- (2) 从 t_1 时刻起,甲方最多还可以发送 5 个数据帧,(1 分)其中第一个帧是 S5,2,(1 分)最后一个数据帧是 S1,2。(1 分)
- (3) 甲方需要重发 3 个数据帧,(1 分)重发的第一个帧是 S2,3。(1 分)
- (4) 甲方可以达到的最大信道利用率是:

$$\frac{7 \times \frac{8 \times 1000}{100 \times 10^6}}{0.96 \times 10^{-3} + 2 \times \frac{8 \times 1000}{100 \times 10^6}} \times 100\% = 50\% (2 \text{ 分})$$

【评分说明】

答案部分正确,酌情给分。