

中国科学院计算技术研究所

一九九二年招收硕士学位研究生入学考试试题

试题名称：软件基础

一. 填空（每空 1 分）

1. 线性表可采用的存储结构有 _____ 和 _____ 二种。
2. 用于拓扑排序的图是 _____ 图。
3. 操作系统的特征是 _____ 和 _____ 。
4. 根据联接在文法符号上的属性之间的依赖关系，可把属性分为 _____ 和 _____ 两大类。
5. 进程间存在的制约关系可以归结为 _____ 和 _____ 两种。
6. 产生死锁的原因是 _____ 和 _____ 。
7. 对逻辑表达式的计算可采用 _____ 和 _____ 方式。
8. 有 n 个结点的完全二叉树的深度为 _____ 。
9. 在串的模式匹配中，若主串长为 m ，子串长为 n ，则 KMP 算法的时间复杂度为 _____ 。
10. 二叉树的遍历方式有 _____、_____ 和 _____ 三种；图的遍历方式有 _____ 和 _____ 两种。

二. 简答

1. （3 分）什么是哈希（Hash）查找？
2. （4 分）UNIX 系统中设备管理的主要特点是什么？
3. （3 分）树(A(B(E(K, L), F), (C(G), D(H(M), I, J))))所对应的二叉树表示是什么？
4. （3 分）已知初始关键字为[49, 38, 65, 97, 76, 13, 27, 49]，现分别用直接插入排序和快速排序方法对之进行排序，请分别给出两种方法第二趟排序之后的结果。
5. （4 分）什么是地址重定位？动态重定位的结果是什么？
6. （3 分）什么是数据的逻辑结构和物理结构？
7. （5 分）程序运行的所需的一片连续存储——活动记录有哪些域？它们各有什么作用？
8. （5 分）有一 C 语言程序如下：

```
main ()
{
    int x;
    x = 25;
    printf ("%d\n", x);
}
```

经过编译联接后，由符号调试器跟踪其执行，却无法检查变量 x 的值，因为调试器报告“无此变量”。试解释其原因。

三. 问答题

1. (8 分) 试比较 Pascal 程序 (语法结构位置任意, 空格等空白字符作为单词分隔符) 和 Fortran 程序 (语法结构位置固定, 空格无意义) 对词法分析的影响。
2. (8 分) 下列文法是否为 LR(1) 文法? 若是, 给出分析表, 若不是, 是说明理由。

$$\begin{array}{lcl} S & \longrightarrow & AA \\ A & \longrightarrow & \mathbf{aA} \mid \mathbf{a} \end{array}$$

3. (8 分) 设一消息缓冲区由如下信息组成:

Sptr	指向发送进程的指针
Nptr	指向下一个消息缓冲区的指针
Size	消息长度
Text	消息正文

请给出两个进程通过该类消息缓冲区进行通讯的过程。

4. (8 分) Swap 指令是一条可在一个存储周期内完成交换两个字节的内容的操作的硬件指令。其功能用 Pascal 语言描述如下:

```

procedure Swap (var a, b: integer)
var temp: integer;
begin
    temp := a;
    a := b;
    b := temp;
end

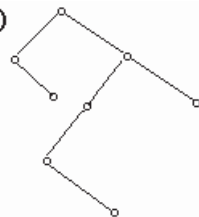
```

请用该指令实现关锁原语 Lock (w) 和开锁原语 Unlock (w)。

四. 算法设计

1. (9分) 二叉树的深度定义为由根结点到叶子结点的最长路径, 设计一算法, 计算二叉树的深度。

(右图二叉树深度为4)



2. (9 分) 写出堆排序 (heap sort) 的算法, 指出其最坏情况下的时间复杂度。

中国科学院计算技术研究所

一九九二年招收硕士学位研究生入学考试试题答案

试题名称：软件基础

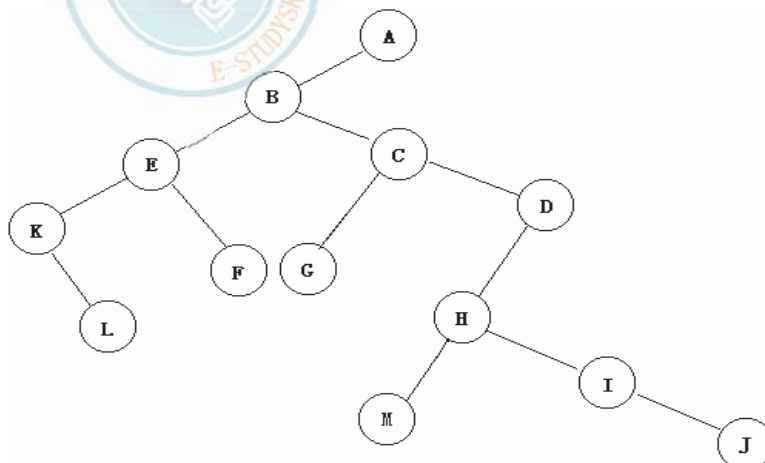
一. 填空

1. 顺序结构、链表结构
2. 有向无环图
3. 程序共行（并发）、资源共享（共享）
4. 综合属性、继承属性
5. 互斥、同步
6. 系统资源不足、进程推进顺序非法
7. 严格计算（直接计算）、短路计算
8. $\lceil \log_2 n \rceil + 1$
9. $O(m+n)$
10. 先序、中序、后序、深度优先、广度优先

二.

1. 由关键字直接计算记录存放位置的查找方法称为哈希查找。（3分）
2. UNIX 系统中设备管理的特点有：（4分）
 - (a) 块设备管理和字符设备管理具有相似的层次结构；
 - (b) 把字符设备作为特别文件处理；
 - (c) 采用了完善的缓冲技术；
 - (d) 引入了“预先读”、“异步写”和“延迟写”方式。

3.



4. 直接插入排序第二趟的结果
[38 49] 65 97 76 13 27 49
快速排序第二趟的结果
[13] 27 [38] 49 [49 65] 76 [97]

5. 由于一个作业装入到与其地址空间不一致的存储空间所引起的、有关地址部分的填整过程，称之为地址重定位。动态重定位的特点是：在程序执行时才进行重定位，它需要硬件的支持。
6. 逻辑结构指数据元素之间的逻辑关系；物理结构指数据结构在计算机存储中的映像。
7. 活动记录包含（答对4个即可得分）
 - (1) 临时工作单元：用于存放计算过程中所需的临时变量；
 - (2) 局部数据：用于存放对应当前活动记录的程序单元的局部数据；
 - (3) 保存机器态部分：用于保存当前的程序计数器、寄存器的值等等；
 - (4) 访问链：把当前执行的程序单元可访问的活动记录联结起来；
 - (5) 控制链：把所有活动记录按它们生成的次序联结起来；
 - (6) 实在参数
 - (7) 返回值
8. 该变量在编译时被优化掉了，所以无法检查其值。

三. 问答题：

1.
 - (a) Fortran 的词法分析比 Pascal 的词法分析复杂；
 - (b) Fortran 语言将一行分为标号、续行、语句、注释四个区，出现在不同区域的符号具有不同的含义。词法分析器必须结合符号出现的位置才可判定其属性；而 Pascal 语言中的记号识别则与位置无关；
 - (c) Fortran 语言认为空格无意义，使得对某些符号的识别取决于后面的符号；而在 Pascal 语言中，每遇过一个空白字符，则认为一个记号已识别完毕。
2. 该文法不是 LR (1) 文法。因为其为二义文法。
$$S \longrightarrow A_1 A_2$$
可将任意输入串 $aa\cdots a$ 截取任意长度的前缀归约为 A_1 ，而将剩下的全部归约为 A_2 。
3.
 - 发送进程在发送消息之前，先在自己的内存空间设置一发送区，填入消息正文等信息，然后调用发送原语 (Send)。进入发送程序后，先找到接受进程的 PCB，再申请获得一消息缓冲区空间，然后互斥进入接受进程的消息队列，找到队尾，把消息缓冲区挂在队尾，把消息正文及长度等从发送区复制到消息缓冲区，最后通过 V 操作通知接收进程，并退出发送程序，发送进程继续执行。
 - 接收进程在读取消息之前，先在自己的内存空间设置一接收区，然后使用 Read 原语，在进入读取程序后，先通过 P 操作检查有无消息，然后互斥进入消息队列，移出队列中的第一个消息，并把消息从缓冲区复制到接收区，最后释放缓冲区，退出读取程序，接收进程继续执

行。

4.

- 锁原语的原定义：

Lock (w):

L: if w = 1 then goto L else w := 1

Unlock (w):

w := 0

- 用 SWAP 指令实现的锁原语：

Lock (w):

key := 1;

repeat

swap (w, key)

until key=0;

Unlock (w)

key := 0;

swap (w, key);

四. 算法设计

1. 设全局变量 n 存放当前树的最大深度，则程序为：

```
procedure visit (lev: integer; root: bitreptr);
begin
  case
    root↑.left = nil and root↑.right = nil:
      if lev > n then n := lev;
      return;
    root↑.left = nil:
      call visit(lev+1; root↑.right);
    root↑.right = nil:
      call visit(lev+1; root↑.left);
    else
      call visit(lev+1; root↑.left);
      call visit(lev+1; root↑.right);
  end /*case*/
end
```

```
begin /* main */
  n := 0;
  call visit(0, root);
  write(n);
end /*main*/
```

2. 堆排序的算法

- (1) 数据结构定义

type rectype = record

```
        key : keytype  
    end;  
    filetype = array [0..n] of rectype;
```

(2) 算法描述

```
procedure heapsort(var r : filetype);  
begin  
    for i := n div 2 downto 1 do    call sift(r , i , n);  
    for i := n downto 2 do  
        [ r[1]  → r[i];  call sift(r , 1 , i-1); ]  
    end; {heapsort}
```

```
procedure sift(var r : filetype; l , m : integer);  
begin  
    i := l; j := 2*i; x := r[i];  
    while j ≤ m do  
        [ if (j < m) and (r[j].key > r[j+1].key) then j := j+1;  
          if x.key > r[j].key  
          then [ r[i] := r[j]; i := j; j := 2*i; ]  
          else j := m+1; ]  
    r[i] := x;  
end; /*sift*/
```

(3) 最坏情况下的时间复杂度为 $O(n \log n)$ 。

