

中国科学院软件所

一九九九年招收硕士学位研究生入学考试试题

试题名称: 软件基础

(编译原理部分)

一、(10分) 文法 G 的产生式如下:

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

(1) 试写出一个语法制导定义, 它输出配对括号个数:

(2) 写一个翻译方案, 打印每个 a 的嵌套深度。如 $((a), a)$, 打印 2 1

缩进属4格而由语法制导输出。

印进属4格而由语法制导输出。

二、(15分) 构造一个 LR(1) 文法 G , 它产生语言 $L(G) = \{w \mid w \in (a \mid b)^*, w$

中 a 和 b 的个数相等}。

$S \rightarrow aBS \mid bAS \mid \epsilon$

$A \rightarrow a \mid bAA$

$B \rightarrow b \mid aBB$

注我认为: $A \rightarrow a \mid bAA$

$B \rightarrow b \mid aBB$

也可以

(操作系统部分)

三、填空 (每空 1 分, 共 10 分. 请将答案写在答题纸上)

1. 程序并发执行与顺序执行时相比产生了一些新特征, 分别是: 间断性、

封闭性、不可再现性

2. 过度地增加多道程序的并行程度, 在内存中会引起 抖动 现象, 反而降低了系统的吞吐量, 理论和实践表明在 (5) 时, CPU 利用得最好。

3. 设备管理中引入缓冲机制的主要原因是为了: 缓和 CPU 与 I/O 设备之间速度不匹配的矛盾, 减小对 CPU 中断频率和 I/O 设备响应时间的限制, 提高 CPU 利用率

4. 在 Unix 等操作系统中, 文件共享有两种方式: (9) 基于索引结点的方式, 利用符号链接实现的链接、(10) 设备之间的并行性

四、(5分) 在一个请求分页系统中, 假如系统分配给一个作业的物理块数为 3; 且此作业的页面走向为 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2。试用 FIFO 和 LRU 两种算法分别计算出程序访问过程中所发生的缺页次数。

五、(10分) 某系统有 r_1, r_2, r_3 三种资源, 在 T_0 时刻 P_1, P_2, P_3, P_4 四个进程对资源的占用和需求情况如下表所示, 此刻系统的可用资源向量为 $(2, 1, 2)$, 问题:

1. 将系统中各种资源总数和此刻各进程对各资源的需求数目用向量或矩阵表示出来;

2. 如果此时 P_1 和 P_2 均发出资源请求向量 $\text{Request}(1, 0, 1)$, 为了保持系统安全性, 应该如何分配资源给这两个进程? 说明你所采用策略的原因;

3. 如果 2 中两个请求立刻得到满足后, 系统此刻是否处于死锁状态。 没有处于死锁状态。


```

    }

    fp2=fopen("stu_Spec","w");
    for (j=0;j<i;j++)
    {
        fprintf(fp2,"%s ", stuSpec[j].num);
        fprintf(fp2,"%s\n",stuSpec[j].spec);
    }

    fclose(fp1);
    fclose(fp2);
}

```

(2)

```

#include <stdio.h>
#include <string.h>

```

```

struct speciality {
    char num[9];
    char spec[21];
    int flag;
} stuSpec[500];

```

main()

```

{
    FILE *fp;

    int i,recs;

    if ((fp=fopen("stu_Spec","r"))==NULL)
    {
        printf("文件打不开");
        return 0 ; // exit(0)
    }

    for ( i=0;i < 500 ;i++)
    {
        if ( fscanf(fp,"%s%s",stuSpec[i].num, stuSpec[i].spec) == EOF )
            break ;
    }

    fclose(fp);
    recs=i;

    for (i=0;i<recs; i++)
    {
        stuSpec[i].flag = 1;

        if (strcmp("9311S009", stuSpec[i].num)==0)
        {
            stuSpec[i].flag = 0;
        }
    }

    fp=fopen("stu_Spec","w");
    for (i=0;i<recs;i++)

```

```
{
    if ( stuSpec[i].flag )
    {
        fprintf(fp,"%s ",stuSpec[i].num );
        fprintf(fp,"%s\n",stuSpec[i].spec );
    }
}
fclose(fp);
}
```

kaoyan.com

资源 进程	Maximum demand			Current allocation		
	r1	r2	r3	r1	r2	r3
P1	3	2	2	1	0	0
P2	6	1	3	4	1	1
P3	3	1	4	2	1	1
P4	4	2	2	0	0	2

(数据结构部分)

六、基本概念 (共 10 分, 请将答案写在答题纸上)

1. 判断正误 (每小题 1 分, 共 5 分. 正确的写“T”, 错误的写“F”)

- (1) 中序遍历一棵二叉排序树的结点就可得到排好序的结点序列。
- (2) 顺序存储方式只能用于存储线性结构。
- (3) 负载因子 (装填因子) 是散列表的一个重要参数, 它反映散列表的装满程度。
- (4) 顺序查找法适用于存储结构为顺序或链接存储的线性表。
- (5) 栈和队列都是限制存取点的线性结构。

2. 选择填空 (5 分)

I. 下图中给出 7 个顶点组成的无向图。从顶点 1 出发, 对它进行深度优先遍历得到的顶点序列是 (1) ; 而进行广度优先遍历得到的顶点序列是 (2)。

(1) A 1354267

B 1347652

C 1534276

D 1247653

E 以上答案均不正确

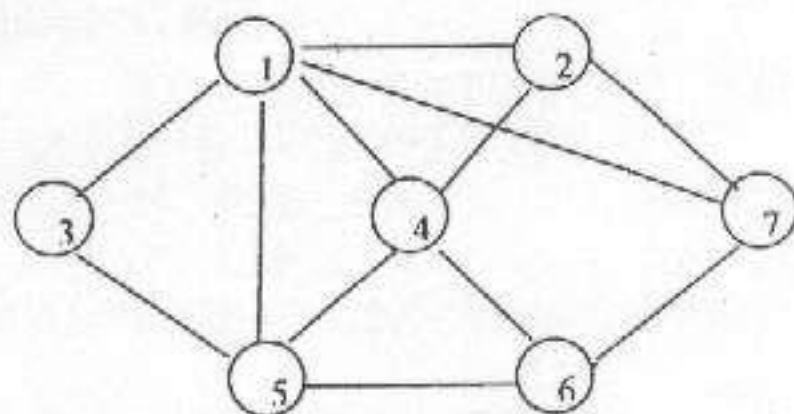
(2) A 1534267

B 1726453

C 1354276

D 1247653

E 以上答案均不正确



题 I 图

II. 从邻接矩阵 $A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ 可以看出, 该图共有 (1) 个顶点。如果是有向图,

该图共有 (2) 条弧; 如果是无向图, 则共有 (3) 条边。

- (1) A 9 B 3 C 6 D 1 E 以上答案均不正确
 (2) A 5 B 4 C 3 D 2 E 以上答案均不正确
 (3) A 5 B 4 C 3 D 2 E 以上答案均不正确

七、算法设计 (共 15 分) (建议用类 PASCAL 语言写出算法)

- (8 分) 二叉排序树采用二叉链表存储。写一个算法, 删除节点值是 X 的结点。要求删除该节点后, 此树仍然是一棵二叉排序树, 并且高度没有增长(注: 可不考虑被删除的结点是根的情况)。
- (7 分) 已知关键字序列 $(K_1, K_2, K_3, \dots, K_{n-1})$ 是大根堆。
 - 试写一算法将 $(K_1, K_2, K_3, \dots, K_{n-1}, K_n)$ 调整为大根堆;
 - 利用 (1) 的算法写一个建大根堆的算法。

(C 语言部分)

八、基础知识 (共 5 分, 请将答案写在答题纸上)

以下是对 C 语言的一些描述, 请选择正确答案。

- 在 C 语言中, 数组名作为参数传递给函数, 作为实在参数的数组名被处理为 ()。
 - 该数组的长度
 - 该数组的元素个数
 - 该数组中各元素的值
 - 该数组的首地址
 - 以上答案均不正确

- 以下程序段的输出是 ()

```
#include <stdio.h>
void fun()
{
    static int a=5;
    a++;
    printf("a=%d\n",a);
}
main()
{
    for (int i=0;i<2;i++)
        fun();
}
```

- A a=6 a=7 B a=5 a=6 C a=1 a=1 D a=5 a=5 E 以上答案均不正确

3 C 语言命令行参数很有特点, 其一般表达形式是 ()。

- A main(int argc, int argv)
- B main(int argc, char argv[])
- C main(int argc, char *argv)
- D main(int argc, char *argv[])
- E 以上答案均不正确

4 类型定义:

```
char S[3]="AB"
```

```
char *P;
```

在执行了语句 $P=S$ 之后, $*(P+2)$ 的值是 ()。

- A 'B'
- B '\0'
- C 不确定
- D 字符 'B' 的地址
- E 以上答案均不正确

5 第一个主要采用 C 语言编写的操作系统是 ()。

- A MS_DOS
- B MS_WINDOWS
- C IBM370
- D UNIX
- E 以上答案均不正确

九、编程 (共 20 分)

1. (10 分) 有 15 个人围成一圈, 顺序从 1 到 15 编号。从第一个人开始报数, 凡报到 n 的人退出圈子。用 C 语言写出程序, 输入 $n(n \geq 1)$ 的值, 输出最后留在圈子里的人的编号。
2. (10 分) 有一磁盘文件 "gstudent" 内存放研究生(研究生数 < 500)的数据, 包括: 姓名, 学号, 性别, 年龄, 住址, 健康状况, 专业。用 C 语言编写程序, 完成下列功能:
 - (1) 要求将学号, 专业信息单独抽出来另建一个简明的研究生专业文件。
 - (2) 从上题的简明 "研究生专业" 文件中删去一个学号是 "9311S009" 的研究生的专业数据。

试题名称：软件基础

(编译原理部分)

一、(10 分)

(1) 为 S, L 引入属性 h ,

产生式	语义规则
$S \rightarrow (L)$	$S.h := L.h + 1$
$S \rightarrow a$	$S.h := 0$
$L \rightarrow L_1, S$	$L.h := L_1.h + S.h$
$L \rightarrow S$	$L.h := S.h$
$S' \rightarrow S$	$\text{print}(S.h)$

(2) 引入属性 d , (继承属性)

$S' \rightarrow \{S.d := 0;\} S$
 $S \rightarrow ' (' \{L.d := S.d + 1;\}$
 $\quad L$
 $\quad ') '$
 $S \rightarrow a \{ \text{print}(S.d); \}$
 $L \rightarrow \{ L_1.d := L.d; \}$
 $\quad L_1$
 $\quad ' , '$
 $\quad \{ S.d := L.d; \}$
 $\quad S$
 $L \rightarrow \{ S.d := L.d; \}$
 $\quad S$

二、(15 分)

文法 G , 产生式如下:

$S \rightarrow aBS \mid bAS \mid \epsilon$

$A \rightarrow a \mid bAA$

$B \rightarrow b \mid aBB$

(操作系统部分)

三、填空(每空 1 分)

- (1)~(3): 中断性; 失去封闭性; 不可再现性
- (4)抖动, (5)产生缺页的平均时间等于系统处理进程缺页的平均时间
- (6)~(8): 缓和 CPU 与 I/O 设备间速度不匹配的矛盾; 减少对 CPU 的中断频率和放宽对中断响应时间的限制; 提高 CPU 和 I/O 设备之间的并行性
- (9)~(10): 基于索引结点的共享方式; 利用符号链实现的共享方式

四、(5 分)

FIFO: 6 次; LRU: 4 次, 并参照实际的运算过程

五、(10 分)

- 系统资源总数向量为: (9, 3, 6)

各进程对资源需求矩阵为:

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 0 & 2 \\ 1 & 0 & 3 \\ 4 & 2 & 0 \end{bmatrix}$$

2. 采用银行家算法进行计算分析可知:

系统可以满足 P2 进程对资源的请求, 将资源分配给 P2 之后, 至少可以找到一个安全的执行序列, 如(P2, P1, P3, P4)使各进程正常运行终结。

系统此刻不能将资源分配给 P1, 虽然可利用资源还可以满足 P1 现在的需求, 但是一旦分配给 P1 之后, 就找不到一个安全执行的序列保证各进程能够正常运行终结, 所以, 进程 P1 应该进入阻塞状态。

3. 系统满足 P1, P2 的请求后, 并没有立即进入死锁状态, 因为这时所有进程没有提出新的资源申请, 全部进程均没有因资源没得到满足而进入阻塞状态。

(数据结构部分)

六、基本概念 (每小题 1 分, 共 10 分),

1 (1) T (2) F (3) T (4) T (5) T

2 I (1) C (2) C II (1) B (2) B (3) D

七、算法设计 (共 15 分)

1 (8 分)

{查找值是 X 的结点}

function BSTSEARCH(var t:bstptr; var f:bstptr; X:keytype):boolean;

begin

f:=t;

while t<>nil do

do case

case t^.key=X

return(true)

case t^.key>X

f:=t;

t:=t^.lchild

case t^.key<X

f:=t;

t:=t^.rchild

endcase

endwhile;

return(false)

end;

{删除值是 X 的结点}

procedure DELNODE(var t:bstptr; X:keytype)

begin

p, f: bstptr;

{p 为叶子结点}

if BSTSEARCH(p, f, X)=TRUE then

if (p = f) then

return

end if

if (p^.lchild=nil) AND (p^.rchild=nil) then


```

    if f^.lchild=p then f^.lchild:=nil
    else f^.rchild:=nil;

    endif
    dispose(p);
    return
end

```

{p 没有左子树的情况}

```

if (p^.lchild=nil) then
    if f^.lchild=p then f^.lchild:=p^.rchild
    else f^.rchild:=p^.rchild;

    endif
    dispose(p);
    return
endif

```

{p 没有右子树的情况}

```

if (p^.rchild=nil) then
    if f^.lchild=p then f^.lchild:=p^.lchild
    else f^.rchild:=p^.lchild

    endif
    dispose(p)
    return
endif

```

{p 既有左子树又有右子树的情况}

```

q:=p; s:=p^.lchild;
{查找 p 的中序直接前驱 s}
while s^.rchild<>NIL DO
    q:=s; s:=s^.rchild;
endwhile
{用 s 的值代替 p 的结点值}
p^.data:=s^.data;
{删除 s}
if q<>p then q^.rchild:=s^.lchild
else q^.lchild:=s^.lchild;

endif
dispose(s);
return

```

```

else
    ERROR("没有找到该结点");
end if

```

end

2 (7 分)

(1) procedure APPEND(VAR k:ARRAY[1..maxsize] OF keytype;
n:integer)

begin

x:keytype;
i,n,j:integer;

{此处假设 "[]" 是向下取整}

x:=k[n]; i:= [n/2]; j:=n;
while x>k[i] AND i>=1 do

```

        k[j]:=k[i];
        j:=i;
        i:=i/2;
    endwhile
    k[j]:=x;
end
(4 分)
(2) procedure BUILDHEAT(var k:filetype;n:integer)
begin
    for i:=2 to n do
        APPEND(k,i)
    endfor
end
(3 分)

```

(C 语言部分)

八、基础知识 (每空 1 分, 共 5 分)

1 D 2 A 3 D 4 B 5 D

九、编程 (共 20 分)

1. (10 分)

```

#include <stdio.h>

#define PEOPLENUM 15

struct person
{
    int number;
    int nextp;
} link[PEOPLENUM+1];

main()
{
    int n;
    int i,count,h;

    scanf("%d",&n);
    if (n<1)
    {
        printf("n 的值必须>0");
        return;
    }

    for (i=1;i<=PEOPLENUM;i++)
    {
        if (i==PEOPLENUM)
            link[i].nextp=1;
        else
            link[i].nextp=i+1;
        link[i].number=i;
    }
    printf("\n");
    count=0;
    h=PEOPLENUM;
    while (count<PEOPLENUM-1)
    {
        i=0;

```



```

while (i!=n)
{
    h=link[h].nextp;
    if (link[h].number) i++;
}
link[h].number=0;
count++;
}
printf("\n 最后留在网中人员的序号是: ");
for (i=1;i<=PEOPLENUM;i++)
    if (link[i].number)
        printf("%6d\n",link[i].number);
}

```

2. (10 分)

(1)

```
#include <stdio.h>
```

```
#include <string.h>
```

```

struct gstudent {
    char name[9];
    char num[9];
    char sex[3];
    int age;
    char addr[30];
    char state[9];
    char spec[21];
} gStudent[500];

```

```

struct speciality {
    char num[9];
    char spec[21];
} stuSpec[500];

```

```
main()
```

```

{
    FILE *fp1, *fp2;
    int i,j;
    if ((fp1=fopen("gstudent","r"))==NULL)
    {
        printf("文件打不开");
        return 0;
    }
    for (i=0; i < 500 ; i++)
    {
        if ( fscanf(fp1,"%s%s%s%0d%s%s%s",
                    gStudent[i].name,gStudent[i].num,gStudent[i].sex,
                    &(gStudent[i].age),
                    gStudent[i].addr,gStudent[i].state,gStudent[i].spec)
            != EOF)
        {
            strcpy( stuSpec[i].num, gStudent[i].num);
            strcpy( stuSpec[i].spec, gStudent[i].spec);
        }
        else break;
    }
}

```