

中国科学院软件研究所

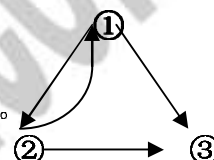
一九九四年招收硕士学位研究生入学考试试题

试题名称：软件基础

操作系统部分（30 分）

一. 填充（1 分×14）

1. 采用单级文件目录的主要缺陷是存在 _____ 问题。
2. 在单道程序运行环境下，常用的作业调度算法有 _____、_____ 和 _____。
3. 特权指令是只能由 _____ 使用的指令。
4. 存储器的保护机制（硬件）有 _____ 保护和 _____ 保护。
5. 预防死锁中的预先分配法和标准分配法，它们分别破坏了产生死锁必要条件中的 _____ 条件和 _____ 条件。
6. 在段式虚拟存储管理中，段表设置“改变位”的目的是为了 _____。
7. 进程有三种基本状态，即① _____ 状态，
② _____ 状态，③ _____ 状态。
当进程由①演变为②或③时，就会立即引起 _____。

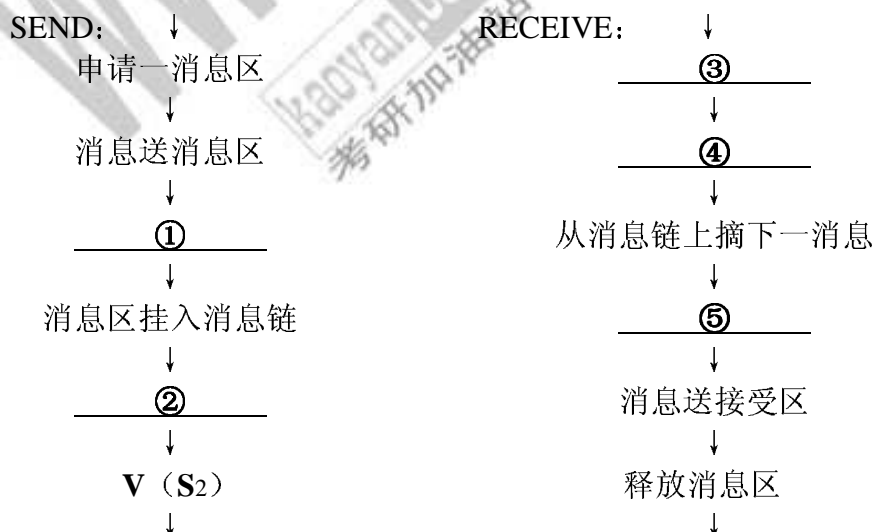


二. 判断（1 分×5，正确的打“√”，错误的打“×”）

1. () 有了动态重定位机构，作业地址空间的代码就可以原封不动地装入到给定的内存中。
2. () 任一时刻，若有执行状态的进程，就一定有就绪状态的进程。
3. () 文件系统中，设置 OPEN 操作的目的是为了将文件复制到内存中。
4. () 临界段是不可中断的程序。
5. () 作业由提交状态进入后备状态的过程是由作业调度程序完成的。

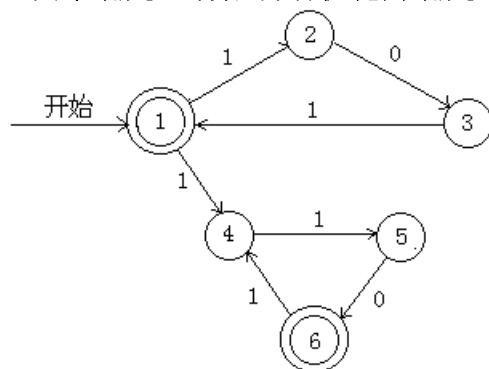
三.（5 分）分页式存储管理与分段式存储管理的主要区别是什么？

四.（6 分）以下是高级通讯原语 SEND 和 RECEIVE 不完整的框图。请填充以适当的 P, V 操作，并说明所用信号量的意义和初值。



语言与编译部分 (35 分)

一. (7 分) 把下面不确定的有限自动机化为确定的有限自动机。



二. (8 分) 有文法:

$$S \longrightarrow (L) \mid a$$

$$L \longrightarrow L, S \mid S$$

给此文法配上语义动作子程序 (或者说为此文法写一个语法制导定义), 它输出配对的括号的个数, 如对于句子 $(a, (a, a))$ 输出是 2。

三. (15 分) 为语言 $\{ a^m b^n \mid n > m \geq 0 \}$ 写三个文法, 它们分别是二义文法, LR(1) 文法和非 LR(1) 且非二义的文法。不必证明所写文法的正确性, 但每个文法的产生式不能超过 4 个。

四. (5 分) 右边是一个 FORTRAN 77 程序。
按语言的语义, 程序的输出结果是什么?
在静态存储分配情况下, 实际的输出结果是什么? 两者是否有区别? 说明理由。

```

CALL SUB
CALL SUB
END
SUBROUTINE SUB
DATA I/10/
WRITE (*, *) I
I=100
END
  
```

程序设计与数据结构部分 (35 分)

一. (8 分) 下面的程序段是合并两条链 (f 和 g) 为一条链 f 的过程。作为参数的两条链都是按节点上 number 值由大到小链接的。合并后新链仍按此方式链接。请填写下述空框, 使程序正确工作。

```

type pointer = ↑ node;
node = record
    number: integer;
    next : pointer
end;

procedure combine (var f: pointer, g: pointer);
var h, p: pointer;
begin new (h);      h↑.next = nil;
    p = h;
  
```

```

while (f<>nil) and (g<>nil) do
  if f↑.number>=g↑.number
  then begin
    p↑.next:= A ; p:= B ; C
  end
  else begin
    p↑.next:= D ; p:= E ; F
  end;
  if f=nil then G ;
  if g=nil then H ;
  f:=h↑.next; dispose(h);

```

二. (12 分) 如果一个数列中的某一段 (至少有两个元素) 的各元素值均相同, 则称之为等值数列段。等值数列段中元素的个数叫做等值数列段的长度。

现有由 N 个元素组成的整数数列 A , 编一程序求 A 中长度最大的所有等值数列段的始末位置。如果没有等值数列段, 则输出特殊标志。

三. (15 分) 编一个程序, 对输入的任意正整数 n , 打印出集合 $\{0, 1, \dots, n-1\}$ 的所有子集。例如: 输入为 3 时, 输出是

```

{}
{0}
{1}
{0, 1}
{2}
{0, 2}
{1, 2}
{0, 1, 2}

```

中国科学院软件研究所

一九九四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

操作系统部分：

一. 填空

1. 重名。
2. 先进先出，最短作业优先，最高响应比优先。
3. 操作系统。
4. 界地址，存储键。
5. 保持和请求，环路。
6. 避免将被淘汰段不必要地写回外存，减少信息传输量。
7. 执行，就绪，封锁（阻塞），进程调度。

二. 判断

1. \checkmark 2. \times 3. \times 4. \times 5. \times

三.

	分页式	分段式
1	单一连续地址空间	三维地址空间
2	页是信息的物理单位	段是信息的逻辑单位
3	页的大小固定，由系统划分，对用户透明	段长度不定，且可变，用户可见
4	（不具有右边特点）	便于动态连接、存储保护，便于增长、修改和共享
5	分配页面大小的空间	分配段大小的空间，为此采用拼接技术

四. ① $P(S_1)$ ② $V(S_1)$ ③ $P(S_2)$

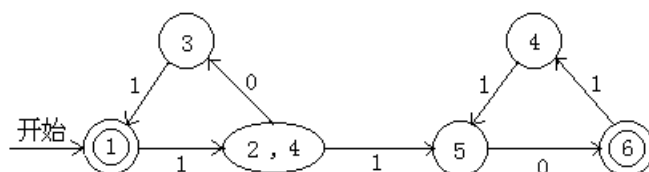
④ $P(S_1)$ ⑤ $V(S_1)$

S_1 是（消息链）互斥信号量，初值为 1。

S_2 是（记录消息个数）同步信号量，初值为 0。

语言与编译部分：


一.




二. 拓广文法, 加入新开始符号 S' 和产生式 $S' \rightarrow S$ 。

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

三. 必须有 $S \rightarrow aSb$ 型的产生式, 以保证 b 的个数不少于 a 的个数。还要有 $S \rightarrow Sb$ 或 $S \rightarrow bS$ 型的产生式, 以保证 b 的个数多余 a 的个数。句子前缀中的 a 和后缀中的 b 的配对方式不同导致不同性质的文法。

二义文法: $S \rightarrow aSb | Sb | b$ 例:  \leftarrow 一种方式
 \leftarrow 另一种方式

LR(1) 文法: $S \rightarrow Sb | S'b$
 $S' \rightarrow aS'b | \epsilon$ 例:  \leftarrow 取前面的 b 与 a 配对

非 LR(1) 且非二义文法: $S \rightarrow aSb | S'$
 $S' \rightarrow S'b | b$ 例:  \leftarrow 取后面的 b 与 a 配对

\uparrow 若只向前看一个符号, 不知后面还有多少个 b 。因此不能确定是做 S' 到 S 的归约呢, 还是移进 b 再做 $S'b$ 到 S' 的归约。

四. 按 FORTRAN77 的语义, 子程序 SUB 的第一次执行输出 10, 第二次输出的值不确定。FORTRAN 语言的实现一般都采用静态存储分配, 变量和存储单元的结合 (binding) 是静态决定的, 在程序运行期间不会改变, 因此 SUB 的两次执行的输出分别是 10, 100。

程序设计和数据结构部分

一. A: f	E: $p \uparrow.\text{next}$
B: $p \uparrow.\text{next}$	F: $g := g \uparrow.\text{next}$
C: $f := f \uparrow.\text{next}$	G: $p \uparrow.\text{next} := g$
D: g	H: $p \uparrow.\text{next} := f$

二. program MAXL (input, output)

```

var len, m, n, l, j: integer;
    A: array[1..100] of integer; //假定 N 为 100

begin
  for j:=1 to 100 do read(A[j]);
  len:=1; l:=1;
  for m:=2 to 100 do
    begin
      if A[m] = A[m-1] then l:=l+1 else l:=1;
      if l>len then len:=l;
    end;
  end;

```

```

    if len=1 then writeln('not have')
    else begin
        l:=1;
        for m:=2 to 100 do
            begin
                if A[m] = A[m-1] then l:=l+1 else l:=1;
                if l=len then
                    begin
                        n:=m+l-l;
                        write(n,m)
                    end
                end
            end
        end
    end.

```

三. program powerset(input,output);

```

var m,n,i,j,k : integer;
begin
    read(n);
    m:=1;
    for i:=1 to n do m:=m*2;
    m:=m-1;          // m= 2n -1
    for i:=0 to m do
        begin
            write('{');
            j:=i; k:=0;
            while (j<>0) do
                begin
                    if j mod 2 <>0 then write(k:5);
                    k:=k+1;
                    j:=j div 2
                end;          //数据间没有逗号
            writeln('}')
        end
    end.

```

若明白下面的对应关系，程序不难编出：

二进制：	0	十进制：	0	集合：	{}
	1		1		{0}
	10		2		{1}
	11		3		{0, 1}
	100		4		{2}
	101		5		{0, 2}
	110		6		{1, 2}
	111		7		{0, 1, 2}