

第 2 章 PL/0 编译程序的实现

第 1 题

PL/0 语言允许过程嵌套定义和递归调用，试问它的编译程序如何解决运行时的存储管理。

答案：

PL/0 语言允许过程嵌套定义和递归调用，它的编译程序在运行时采用了栈式动态存储管理。（数组 CODE 存放的只读目标程序，它在运行时不改变。）运行时的数据区 S 是由解释程序定义的一维整型数组，解释执行时对数据空间 S 的管理遵循后进先出规则，当每个过程(包括主程序)被调用时，才分配数据空间，退出过程时，则所分配的数据空间被释放。应用动态链和静态链的方式分别解决递归调用和非局部变量的引用问题。

第 2 题

若 PL/0 编译程序运行时的存储分配策略采用栈式动态分配，并用动态链和静态链的方式分别解决递归调用和非局部变量的引用问题，试写出下列程序执行到赋值语句 $b := 10$ 时运行栈的布局示意图。

```
var x,y;  
procedure p;  
  var a;  
  procedure q;  
    var b;  
    begin (q)  
      b := 10;  
    end (q);  
  procedure s;  
    var c,d;  
    procedure r;  
      var e,f;  
      begin (r)  
        call q;  
      end (r);  
    begin (s)  
      call r;  
    end (s);  
  begin (p)  
    call s;
```

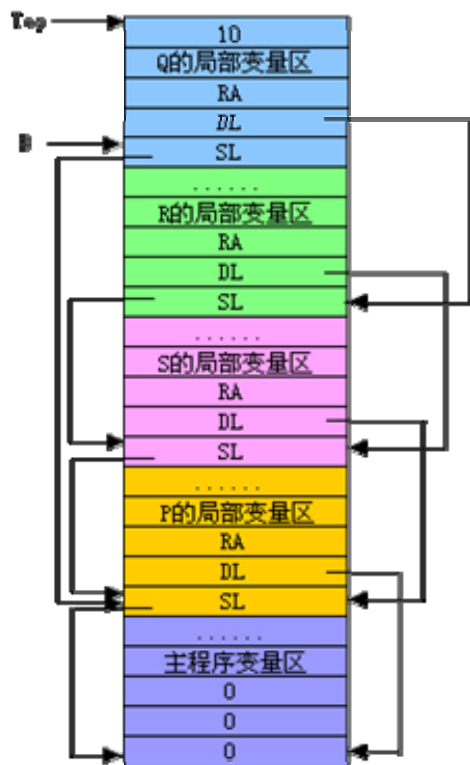
```

end (p);
begin (main)
  call p;
end (main).

```

答案:

程序执行到赋值语句 $b := 10$ 时运行栈的布局示意图为:



第 3 题

写出题 2 中当程序编译到 r 的过程体时的名字表 $table$ 的内容。

name	kind	level/val	adr	size

答案:

题 2 中当程序编译到 r 的过程体时的名字表 $table$ 的内容为:

name	kind	level/val	adr	size
x	variable	0	dx	
y	variable	0	dx+1	
p	procedure	0	过程 p 的入口 (待填)	5

a	variable	1	dx	
q	procedure	1	过程 q 的入口	4
s	procedure	1	过程 s 的入口（待填）	5
c	variable	2	dx	
d	variable	2	dx	
r	procedure	2	过程 r 的入口	5
e	variable	3	dx	
f	variable	3	dx+1	

注意：q 和 s 是并列的过程，所以 q 定义的变量 b 被覆盖。

第 4 题

指出栈顶指针 T，最新活动记录基地址指针 B，动态链指针 DL，静态链指针 SL 与返回地址 RA 的用途。

答案：

栈顶指针 T，最新活动记录基地址指针 B，动态链指针 DL，静态链指针 SL 与返回地址 RA 的用途说明如下：

T： 栈顶寄存器 T 指出了当前栈中最新分配的单元(T 也是数组 S 的下标)。

B： 基址寄存器，指向每个过程被调用时，在数据区 S 中给它分配的数据段起始地址，也称基地址。

SL： 静态链，指向定义该过程的直接外过程（或主程序）运行时最新数据段的基地址，用以引用非局部（包围它的过程）变量时，寻找该变量的地址。

DL： 动态链，指向调用该过程前正在运行过程的数据段基地址，用以过程执行结束释放数据空间时，恢复调用该过程前运行栈的状态。

RA： 返回地址，记录调用该过程时目标程序的断点，即调用过程指令的下一条指令的地址，用以过程执行结束后返回调用过程时的下一条指令继续执行。

在每个过程被调用时在栈顶分配 3 个联系单元，用以存放 SL，DL，RA。

第 5 题

PL/0 编译程序所产生的目标代码是一种假想栈式计算机的汇编语言，请说明该汇编语言中下列指令各自的功能和所完成的操作。

(1) INT 0 A

(2) OPR 0 0

(3) CAL LA

答案：

PL/0 编译程序所产生的目标代码中有 3 条非常重要的特殊指令，这 3 条指令在 code 中的位置和功能以及所完成的操作说明如下：

INT 0 A

在过程目标程序的入口处，开辟 A 个单元的数据段。A 为局部变量的个数+3。

OPR 0 0

在过程目标程序的出口处，释放数据段（退栈），恢复调用该过程前正在运行的过程的数据段基址寄存器 B 和栈顶寄存器 T 的值，并将返回地址送到指令地址寄存器 P 中，以使调用前的程序从断点开始继续执行。

CALL A

调用过程，完成填写静态链、动态链、返回地址，给出被调用过程的基址值，送入基址寄存器 B 中，目标程序的入口地址 A 的值送指令地址寄存器 P 中，使指令从 A 开始执行。

第 6 题

给出对 PL/0 语言作如下功能扩充时的语法图和 EBNF 的语法描述。

(1) 扩充条件语句的功能使其为：

if 〈条件〉 **then** 〈语句〉 [**else** 〈语句〉]

(2) 扩充 repeat 语句为：

repeat 〈语句〉 {; 〈语句〉} **until** 〈条件〉

答案：

对 PL/0 语言作如下功能扩充时的语法图和 EBNF 的语法描述如下：

(1) 扩充条件语句的语法图为：



EBNF 的语法描述为：〈条件语句〉 ::= if 〈条件〉 then 〈语句〉 [else 〈语句〉]

(2) 扩充 repeat 语句的语法图为：



EBNF 的语法描述为：〈重复语句〉 ::= repeat 〈语句〉 {; 〈语句〉} until 〈条件〉