

COMPILADORES

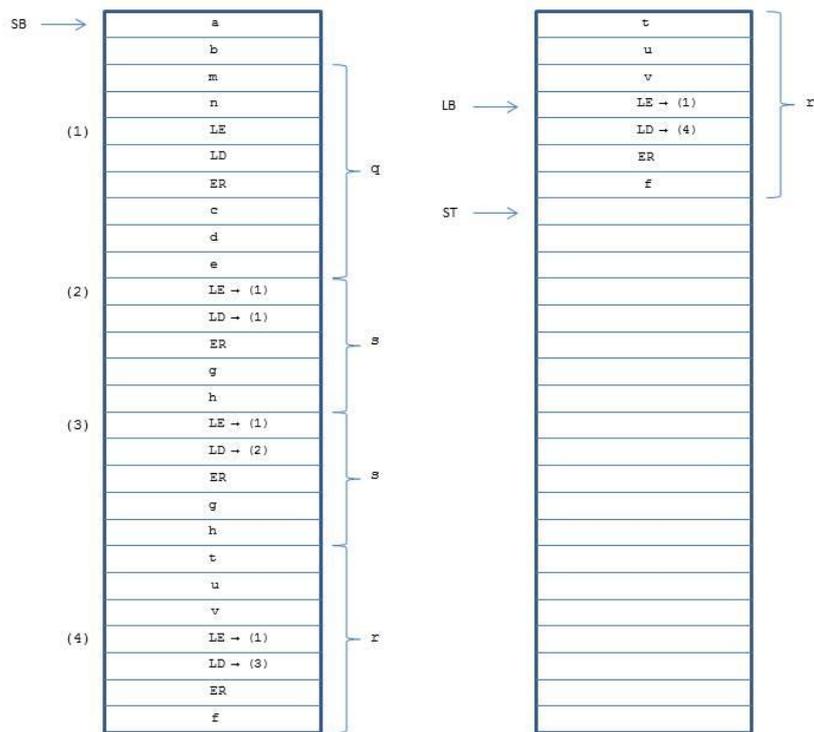
Prova 2 – 28/04/2017 – Prof. Marcus Ramos

Considere o esboço de programa Pascal abaixo e a seguinte seqüência de execução:

$p \rightarrow q \rightarrow s \rightarrow s \rightarrow r \rightarrow r$.

```
program p;  
  var a,b: integer;  
  procedure q (m,n: integer);  
    var c,d,e: integer;  
    procedure r (t,u,v : integer);  
      var f: integer;  
      begin (* r *)  
        ...  
      end; (* r *)  
    procedure s;  
      var g,h: integer;  
      begin (* s *)  
        ...  
      end; (* s *)  
    begin (* q *)  
      ...  
    end; (* q *)  
  begin (* p *)  
    ...  
  end; (* p *)
```

Questão 1 (2,0 pontos): Mostre a situação da pilha de execução com todos os frames, variáveis, parâmetros e links (estático e dinâmico) corretamente identificados e posicionados durante a execução do último bloco da seqüência.



Questão 2 (2,0 pontos): Determine o escopo de cada nome deste programa (variáveis, parâmetros e blocos).

- a: p
- b: p
- q: p
- m: q
- n: q
- c: q
- d: q
- e: q
- r: q
- t: r
- u: r
- v: r
- f: r
- s: q
- g: s
- h: s

Questão 3 (2,0 pontos): Determine o nível onde cada identificador deste programa está declarado (variáveis, parâmetros e blocos).

a: 0
b: 0
q: 0
m: 1
n: 1
c: 1
d: 1
e: 1
r: 1
t: 2
u: 2
v: 2
f: 2
s: 1
g: 2
h: 2

Questão 4 (2,0 pontos): Determine, na forma de um par deslocamento/registrador, o endereço de cada variável e de cada parâmetro em cada ponto do programa onde ele possa ser referenciado (considere que todos os objetos são alocados em uma única unidade de memória).

No corpo do programa principal:

a: 0 [SB]
b: 1 [SB]

No corpo do procedimento q:

a: 0 [SB]
b: 1 [SB]
m: -2 [LB]
n: -1 [LB]
c: 3 [LB]
d: 4 [LB]
e: 5 [LB]

No corpo do procedimento r:

a: 0 [SB]
b: 1 [SB]
m: -2 [L1]
n: -1 [L1]
c: 3 [L1]
d: 4 [L1]
e: 5 [L1]
t: -3 [LB]
u: -2 [LB]

v: -1 [LB]
f: 3 [LB]

No corpo do procedimento s:

a: 0 [SB]
b: 1 [SB]
m: -2 [L1]
n: -1 [L1]
c: 3 [L1]
d: 4 [L1]
e: 5 [L1]
g: 3 [LB]
h: 4 [LB]

Questão 5 (2,0 pontos): Suponha que o trecho de código apresentado a seguir esteja contido no bloco r. Considere os templates de código estudados em sala de aula e mostre o código gerado para a máquina de pilha TAM (não é necessário usar os endereços das variáveis):

```
while (a<c+f) do
    if (m<>t) then repeat b:=b+u; f:=f-1; until (f==0)
    else (if v+3==d+e*2 then a:=a+c+v+f);
```

```
L_0001:    LOAD a
           LOAD c
           LOAD f
           CALL add
           CALL lt
           JUMPIF (0) L_0002
           LOAD m
           LOAD t
           CALL neq
           JUMPIF (0) L_0003
L_0004:    LOAD b
           LOAD u
           CALL add
           STORE b
           LOAD f
           LOADL 1
           CALL sub
           STORE f
           LOAD f
           LOADL 0
           CALL eq
           JUMPIF (0) L_0004
           JUMP L_0006
L_0003:    LOAD v
           LOADL 3
```

```
CALL add
LOAD d
LOAD e
LOADL 2
CALL mult
CALL add
CALL eq
JUMPIF (0) L_0005
LOAD a
LOAD c
LOAD v
LOAD f
CALL add
CALL add
CALL add
STORE a
L_0005:
L_0006:
JUMP L_0001
L_0002:
```