

Linguagens de Programação

Âmbito, Funções e Gestão de memória

1. Considere o seguinte fragmento de código C:

```
1  {
2      int a = 5, b = 2, c = 0;
3      while (b > 0)
4      {
5          int a = 3;
6          a = (a - b) * (a + b);
7          c += a; --b;
8      }
9      {
10         int d;
11         {
12             int a = 15, b;
13             b = c + d + a;
14         }
15         d = b;
16     }
17     return c;
18 }
```

- (a) Identifique os vários blocos do código (nomeie-os A, B, ...).
 - (b) Em cada bloco, diga quais as variáveis locais e globais que nele ocorrem.
 - (c) Para cada variável global de cada bloco, calcule a distância (em número de blocos) entre o bloco onde ela está a ser usada e o bloco onde ela é declarada.
 - (d) Desenhe o registo de activação de cada bloco. Assuma que a unidade de memória é a palavra e que cada inteiro e cada apontador ocupa uma palavra.
 - (e) Estude a evolução da pilha de execução ao longo da execução do código apresentado.
2. Calcule o resultado do programa seguinte, escrito numa notação à la Algol/Pascal, para os modos de passagem de parâmetros indicados.

```

begin
  integer i;
  procedure pass(x, y: integer)
  begin
    x := x + 1;
    y := x + 1;
    x := y;
    i := i + 1
  end;
  i := 1;
  pass(i, i);
  print i
end

```

- (a) Passagem de parâmetros por valor.
- (b) Passagem de parâmetros por referência.
- (c) Passagem de parâmetros por valor-resultado. O modo de passagem de argumentos por valor-resultado (ou *copy-in/copy-out*) funciona como a passagem de argumentos por valor até ao fim da execução do corpo da função ou procedimento. Nesta altura, e antes da função ou procedimento retornar, o valor dos argumentos formais (*formal arguments*) é copiado para os argumentos efectivos (*actual arguments*).

3. Considere o código seguinte, escrito numa notação à la Algol/Pascal:

```

var x: integer;
procedure p(y: integer)
begin
  y := 1;
  x := 10
end;
x := 0;
p(x);

```

Traduza o código apresentado para ML, de modo a reflectir os modos de passagem de argumentos seguintes, e indique os diversos valores que `x` tem ao longo da execução do programa:

- (a) Passagem de parâmetros por valor.
- (b) Passagem de parâmetros por referência.

(c) Passagem de parâmetros por valor-resultado.

4. [Mitchell 7.8] Para o programa ML seguinte estude a evolução do conteúdo da pilha de execução durante a sua avaliação, e determine o valor das ocorrências de x (nas linhas 2, 4 e 5) e o valor final, quando

(a) o âmbito dos identificadores é estático (*lexical/static scope*).

(b) o âmbito dos identificadores é dinâmico (*dynamic scope*).

```
1  let val x = 2 in
2      let fun f y = x + y in
3          let val x = 7 in
4              x +
5              f x
6          end
7      end
8  end;
```

5. [Mitchell 7.12] Para o seguinte programa

```
1  val x=5;
2  fun f(y)=(x+y)-2;
3  fun g(h)=let val x=7 in h(x) end;
4  let val x=10 in g(f) end;
```

(a) Represente os registos de activação presentes no stack de execução após a chamada de h no corpo de g .

(b) Qual o valor desta expressão? Porquê?

6. [Mitchell 7.13] Para o seguinte programa

```
1  val x=5;
2  fun f(s)=
3      let val z =[1,2,3]
4          fun g(w)=w+x+s
5          in
6              g
7          end;
8  val h=
9      let val x=7
10      in
11          f(3) end;
12  h(2);
```

- (a) Escreva o tipo de cada um dos identificadores declarados (**x**, **f** e **h**).
- (b) Represente os registos de activação presentes no *stack* de execução após a chamada de **h** no final deste segmento de código (assuma que os RA serão retirados da memória por um *garbage collector*).
- (c) Qual o valor desta expressão? Explique quais os valores que são adicionados e porquê.

7. [Mitchell 7.14] Para o seguinte programa

```

1  fun myop(x,y)=x*y;
2  fun recurse(i)=
3      if i=0 then 1
4      else myop(i,recurse(i-1));
5  let
6      fun myop(x,y)=x+y
7  in
8      recurse(1)
9  end;
```

- (a) Represente os registos de activação presentes no *stack* de execução após a última chamada a **myop** na linha 4 deste fragmento de código.
- (b) Diga qual o valor desta expressão, explicando como o registo de activação desenhado permite encontrar que valor de **myop** usar
 - i. em âmbito estático
 - ii. em âmbito dinâmico