

Apuntes

Clase 6-Binding y Registros de Activación

Clase 7-Clasificación de Variables

BINDING**1. Clasifique en semántica, pragmática y sintáctica cada una de las siguientes afirmaciones:**

- El lenguaje C permite pasaje de parámetros por referencia
- El compilador ANSI C produce ejecutables mas lentos que el mismo de Borland.
- La potencia en TURBO PASCAL se indica con el símbolo “^” entre la base y el exponente.
- En el lenguaje Fortran no se puede escribir más de una sentencia por renglón.
- En PL/I las variables se pueden declara antes o después de su primer uso.
- Las variables long double de C++ de Linux son tan rápidas como las double
- Las variables de Basic son todas globales.

2. Sea el siguiente fragmento de programa, en un lenguaje determinado:

```
var1 = 205
var2 = 10
var3 = 1024
var2 = 205
var1= “diez”
```

- a) ¿Qué puede deducir del lenguaje en que se han escrito las instrucciones anteriores?
- b) ¿Qué ocurre si se ejecuta $\text{var1} = \text{var2} + \text{var3}$?

3. El siguiente programa:

```
Y = 0
X = 0
Z = 255
REPEAT
    Z= Z-1
UNTIL (Z<0)
```

Tarda mucho más que

```
Z= 255
Y= 0
X= 0
REPEAT
    Z= Z-1
UNTIL (Z<0)
```

Justifique esta afirmación.

¿Qué puede deducir del lenguaje?

4. De acuerdo al siguiente segmento de código:

```

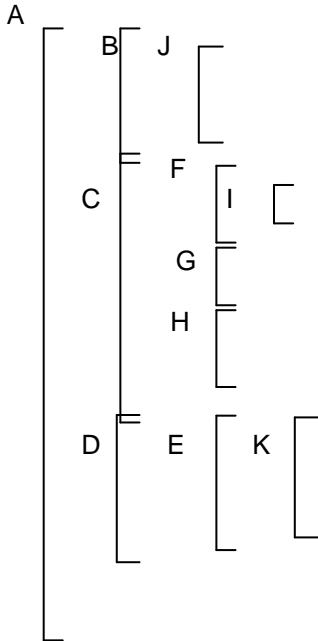
Prog A
  var a: integer
  Prog B
    var b: integer
    Prog C
      var c,d: integer
      -----→ (1)
      Prog D
        var d: integer
        -----→ (2)
        Prog E
          -
          End E
        -
        End D
      -
      End C
    -
    End B
  Prog F
    var f: float
    -----→ (3)
    Prog H
      -
      End H
    -
    End F
  Prog G
    Prog I
      var l: float
      -----→ (4)
      Prog J
        -
        End J
      -
      End I
    -
    End G
  -
  End A
    
```

Asumiendo alcance estático:

1. Indicar quién puede llamar a quién.
2. Construir el árbol de anidamiento del programa.
3. Hacer un listado completo de todos los posibles llamados en los que el puntero de la cadena estática coincide con el puntero de la cadena dinámica.
4. Cuáles son las variables visibles en los puntos indicados

NOTA: Se entiende por llamado: A->B (un solo llamado)
 Se entiende por secuencia de llamados: A->B->C (mas de un llamado)

5. Sea un programa cuyo esquema de bloques es el siguiente:



¿ Cuáles de las siguientes secuencias son posibles y cuáles no?. Marcar el llamado incorrecto en los casos que corresponda.

En todas las secuencias construir la pila de ejecución incluyendo la cadena estática.

- a) $A \rightarrow B \rightarrow J \rightarrow A \rightarrow C \rightarrow G \rightarrow A \rightarrow D \rightarrow E$
- b) $A \rightarrow D \rightarrow E \rightarrow C \rightarrow G \rightarrow G \rightarrow E$
- c) $A \rightarrow C \rightarrow D \rightarrow D \rightarrow E \rightarrow E \rightarrow K$
- d) $A \rightarrow C \rightarrow H \rightarrow H \rightarrow C \rightarrow A$
- e) $A \rightarrow B \rightarrow J \rightarrow C \rightarrow G \rightarrow F \rightarrow I \rightarrow F$
- f) $A \rightarrow B \rightarrow F \rightarrow I \rightarrow I \rightarrow G \rightarrow H \rightarrow H$

6. Sea el siguiente fragmento de programa en Pascal:

```

program main
  var a,y,c : integer;
  procedure sub1;
    var x,y,z : integer;
    begin {sub1}
      ..
    end {sub1}
  procedure sub2
    var c,x,w: integer;
    procedure sub3;
      var a,d,z: integer;
      begin {sub3}
        ..
      end {sub3}
    begin {sub2}
      ..
    end {sub2}
  begin {main}
  ..
end {main}
  
```

Se deben listar todos los nombre de los **identificadores**, junto al procedimiento correspondiente donde fueron declarados, que se encuentran visibles en los cuerpos de sub1, sub2 y sub3.

7. Considere el siguiente esqueleto de programa:

```

procedure main
  var c,d: integer
  procedure sub1
    var a,b: integer;
    {-----→(1)
    .....
    call sub3;
  }
  procedure sub2
    var b,c: integer;
    {-----→(2)
    .....
    call sub1;
    .....
  }
  procedure sub3
    var b: integer;
    procedure sub4
      var d: integer;
      {-----→ (3)
      d:=b+a;
      }
      {-----→(4)
      .....
      call sub4;
    }
    {-----→(5)
    .....
    call sub2;
  }

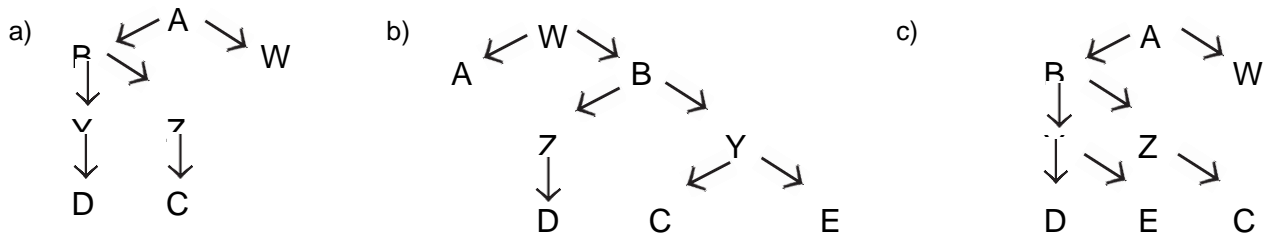
```

Asumiendo alcance estático, mencionar qué variables son visibles en cada punto del programa.

8. Sean A, B, C, D, E, W, Y y Z los nombres de seis unidades de un programa que están anidadas unas en otras. Las llamadas permitidas entre estas unidades están indicadas en la siguiente tabla:

	A	B	C	D	E	W	Y	Z
A	SI	SI	X	X	X	SI	X	X
B	SI	SI	X	X	X	SI	SI	SI
C	SI	SI	SI	X	X	SI	SI	SI
D	SI	SI	X	SI	SI	SI	SI	SI
E	SI	SI	X	SI	SI	SI	SI	SI
W	SI	SI	X	X	X	SI	X	X
Y	SI	SI	X	SI	SI	SI	SI	SI
Z	SI	SI	SI	X	X	SI	SI	SI

Indicar cuál de los siguientes árboles de anidamiento corresponden a la tabla, especificando para el/los correspondientes: llamados recursivos, llamados locales y llamados globales con la distancia correspondiente.



CLASIFICACION DE VARIABLES

9. Se tiene un programa con un procedimiento cuyo registro de activación varía de tamaño en diferentes ejecuciones. ¿Qué puede deducir de este lenguaje?

10. Sea la siguiente matriz entera:

var Z: array [7..45,9..22] of integer;

a) ¿Cuáles de los números 7, 45, 9 y 22 deben almacenarse en el registro de activación? ¿En qué otro lugar puede almacenarse?

b) Ídem si la variable es semidinámica.

11. Sea la matriz del ejercicio anterior. Si cada entero ocupa dos bytes y el comienzo de Z se encuentra en la dirección 3506C, ¿qué componente se encuentra almacenado en la dirección 350B2.

a) Si se encuentra organizada por filas.

b) Si se encuentra organizada por columnas.

12. Indique si las siguientes afirmaciones son verdaderas o falsas. Explicar por qué.

a) El tiempo de vida de una variable declarada en un programa en lenguaje Pascal, sin llamadas a procedimientos ni funciones, coincide siempre con su alcance.

b) Una variable declarada en C ó C++ como *static* posee almacenamiento y alcance estáticos.

c) Si una variable definida en el interior de un bloque de "C" {} contiene una inicialización, por ejemplo, `int x = 3;` la misma se ejecuta cada vez que comienza el bloque.

d) Un bloque anónimo en C genera un nuevo registro de activación.

e) El lenguaje C utiliza descriptores para almacenar los límites de sus vectores.

f) Pascal incorpora chequeo de límites en el código ejecutable de sus programas.

g) ADA y Algol conocen los límites de todos sus vectores en tiempo de compilación.