

Paradigmas de la Programación

Práctico 3: Fundamentos de lenguajes imperativos II: alcance, clausuras y pasaje de parámetros

Laura Alonso Alemany

Ezequiel Orbe

31 de marzo de 2015

Seleccione la(s) respuesta(s) correcta(s)

1. Los *activation records* representan...
 - a) unidades en la pila de ejecución
 - b) *stack frames*
 - c) bloques del texto del programa
 - d) todos los anteriores
 - e) ninguno de los anteriores
2. Todo activation record necesariamente tiene...
 - a) control link
 - b) access link
 - c) variables locales
 - d) todas las anteriores
 - e) ninguna de las anteriores
3. La clausura de una función es necesaria para...
 - a) definir funciones en un alcance global
 - b) mantener la referencia a las variables locales de la función si tenemos alcance dinámico
 - c) mantener la referencia a las variables locales de la función si tenemos alcance estático
 - d) mantener la referencia a las variables libres de la función si tenemos alcance dinámico
 - e) mantener la referencia a las variables libres de la función si tenemos alcance estático
4. Si el resultado del siguiente programa es 19, qué tipo de alcance tiene el lenguaje de programación en el que está escrito?

```
val x = 4;  
  fun f(y) = x*y;  
  fun g(x) = let  
    f(3) + x;  
  g(7);
```

 - a) estático
 - b) dinámico

5. Si el Browse del siguiente programa muestra 4, qué tipo de pasaje de parámetros tiene?

```
declare A=2 B=4 C=1
proc {P X Y Z}
  C=5
  X=Z
  C=4
  Y=Z+A
end
in {P A B A+C}
{Browse B}
```

- a) por valor
- b) por referencia
- c) por valor-resultado
- d) por nombre
- e) por por necesidad

6. Si el Browse del siguiente programa muestra 5, qué tipo de pasaje de parámetros tiene?

```
declare A=2 B=4 C=1
proc {P X Y Z}
  C=5
  X=Z
  C=4
  Y=Z+A
end
in {P A B A+C}
{Browse B}
```

- a) por valor
- b) por referencia
- c) por valor-resultado
- d) por nombre
- e) por por necesidad

7. La ventaja del pasaje de parámetros por referencia es que...

- a) reduce aliasing.
- b) ocupa menos lugar en la memoria.
- c) no tiene efectos fuera de la función llamada.
- d) permite actuar en la función que llama, si el programador lo necesita.

8. La ventaja del pasaje de parámetros por valor es que...

- a) reduce aliasing.
- b) ocupa menos lugar en la memoria.
- c) no tiene efectos fuera de la función llamada.
- d) permite actuar en la función que llama, si el programador lo necesita.

9. La ventaja del pasaje de parámetros por valor-resultado es que...

- a) reduce aliasing.
 - b) ocupa menos lugar en la memoria.
 - c) no tiene efectos fuera de la función llamada.
 - d) permite actuar en la función que llama, si el programador lo necesita.
10. Los métodos de pasaje por parámetros que siguen la filosofía de evaluación perezosa (*lazy evaluation*) son...
- a) pasaje por referencia.
 - b) pasaje por valor.
 - c) pasaje por valor-resultado.
 - d) pasaje por nombre.
 - e) pasaje por necesidad.
11. a)
- b)
 - c)

Ejercicios prácticos

1. Diagrame la pila de ejecución del siguiente programa en Bash, incluyendo la estructura de control links y access links, asumiendo que el lenguaje de programación tiene alcance estático.

```
x=1
function g () { echo $x ; x=2 ; }
function f () { local x=3 ; g ; }
f # does this print 1, or 3?
echo $x
```

En realidad este programa está escrito en Bash, que tiene alcance dinámico. Justifique por qué puede ser que Bash se haya diseñado con lenguaje dinámico.

Qué imprimirá este programa, ahora que sabemos que tiene alcance dinámico?

2. Reescriba el siguiente programa en Python para que sea válido en Perl en modo estricto, poniendo atención al lugar de declaración de la variable, para que su alcance sea exactamente el mismo en ambos programas.

```
if c:
    a = 'foo'
else:
    a = ''
```

3. Reescriba el siguiente programa para que sea un programa en C estándar (sin usar gcc con `-fnested-functions`) con la misma semántica.

```
void f(void)
{
    // Define a function called g
    void g(void)
    {
    }
    // Call g
    g();
}
```

4.

5.