

## Tema 2.7. Subprogramas. Traducción

Pedro Javier Rodríguez Rodrigo, Víctor Cuadrado Juan

6 de mayo de 2014

# Organización de la memoria

- Es posible acceder a los datos globales

# Organización de la memoria

- Es posible acceder a los datos globales
- Desde cualquier registro de activación es necesario referir al registro de activación asociado con el bloque padre (que no tiene porque ser necesariamente el registro de activación anterior)

# Organización de la memoria

- Es posible acceder a los datos globales
- Desde cualquier registro de activación es necesario referir al registro de activación asociado con el bloque padre (que no tiene porque ser necesariamente el registro de activación anterior)
- Dos Posibles organizaciones:

# Organización de la memoria

- Es posible acceder a los datos globales
- Desde cualquier registro de activación es necesario referir al registro de activación asociado con el bloque padre (que no tiene porque ser necesariamente el registro de activación anterior)
- Dos Posibles organizaciones:
  - Enlaces estáticos

# Organización de la memoria

- Es posible acceder a los datos globales
- Desde cualquier registro de activación es necesario referir al registro de activación asociado con el bloque padre (que no tiene porque ser necesariamente el registro de activación anterior)
- Dos Posibles organizaciones:
  - Enlaces estáticos
  - Displays

# Enlaces estáticos

- En el registro de activación se incluye un enlace al registro de activación del bloque padre (enlace estático)
- La memoria se organiza en forma de pila de registros de activación, enlazados a través de los enlaces estáticos

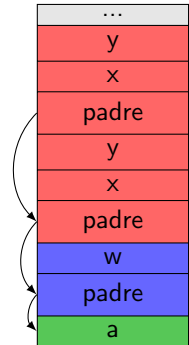
# Enlaces estáticos: Ejemplo

```
proc proc1(){  
    x: num;  
    y: num;  
    proc1();  
}  
  
proc proc2(){  
    w: bool;  
    proc1();  
}  
  
main(){  
    a: bool;  
    proc2();  
}
```

Mem. instrucciones



Regs. de activación





# Enlaces estáticos: Problemas

¿Qué problemas hay?

# Enlaces estáticos: Problemas

¿Qué problemas hay?

- 1 La recuperación del enlace de un identificador global supone seguir toda la cadena de enlaces estáticos. Si el identificador ha sido declarado  $k$  niveles por encima, es necesario realizar  $k$  indirecciones hasta llegar al correspondiente registro de activación

# Enlaces estáticos: Problemas

¿Qué problemas hay?

- 1 La recuperación del enlace de un identificador global supone seguir toda la cadena de enlaces estáticos. Si el identificador ha sido declarado  $k$  niveles por encima, es necesario realizar  $k$  indirecciones hasta llegar al correspondiente registro de activación
- 2 Hay que considerar la complejidad de generar código que gestione de manera adecuada los enlaces estáticos

# Enlaces estáticos: Problemas

¿Qué problemas hay?

- 1 La recuperación del enlace de un identificador global supone seguir toda la cadena de enlaces estáticos. Si el identificador ha sido declarado  $k$  niveles por encima, es necesario realizar  $k$  indirecciones hasta llegar al correspondiente registro de activación
- 2 Hay que considerar la complejidad de generar código que gestione de manera adecuada los enlaces estáticos

Solución:

Almacenar los enlaces estáticos *fuera* de los registros de activación. La estructura que los almacena se llama **display**.

# Display

- Secuencia de celdas consecutivas que apuntan a registros de activación

# Display

- Secuencia de celdas consecutivas que apuntan a registros de activación
- La celda  $i$  apunta al registro de activación que está siendo utilizado en el nivel de anidamiento  $i$

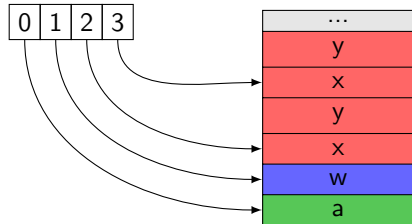
# Display

- Secuencia de celdas consecutivas que apuntan a registros de activación
- La celda  $i$  apunta al registro de activación que está siendo utilizado en el nivel de anidamiento  $i$
- Esta estructura facilita el acceso a los datos globales: el enlace para un identificador declarado en un bloque que se encuentra a profundidad  $i$  estará en el registro de activación referido por la celda  $i$  del display (el *display*  $i$  a partir de ahora)

# Display: Ejemplo

```
proc proc1(){  
    x: num;  
    y: num;  
    proc1();  
}  
proc proc2(){  
    w: bool;  
    proc1();  
}  
main(){  
    a: bool;  
    proc2();  
}
```

Displays y regs. de activación





# Memoria de un programa I

dibujo resumen de toda la memoria del programa (instrucciones, estatico, reg. activacion, heap)

## Memoria de un programa II

Las primeras celdas de la memoria se destinarán a mantener la información de estado necesaria para gestionar adecuadamente la pila de registros de activación:

- Registro *CP*: Contendrá siempre la dirección de la **última celda ocupada** por la pila de registros de activación (cuando la pila esté vacía, el valor de *CP* será la dirección de la celda anterior -la última celda ocupada por el display-)

## Memoria de un programa II

Las primeras celdas de la memoria se destinarán a mantener la información de estado necesaria para gestionar adecuadamente la pila de registros de activación:

- Registro *CP*: Contendrá siempre la dirección de la **última celda ocupada** por la pila de registros de activación (cuando la pila esté vacía, el valor de *CP* será la dirección de la celda anterior -la última celda ocupada por el display-)
- *Display*

# Estructura de los Registros de activación

dibujo del registro de activación

# Inicio

- Se fija el *display 0* a la primera celda de datos estáticos

dibujo

# Inicio

- Se fija el *display 0* a la primera celda de datos estáticos
- Se fija el *CP* a la posición de la última celda del display (la última celda ocupada)

dibujo

# Inicio

- Se fija el *display 0* a la primera celda de datos estáticos
- Se fija el *CP* a la posición de la última celda del display (la última celda ocupada)
- Con ello se consigue un esquema homogéneo de direccionamiento de datos estáticos y de datos en los registros de activación

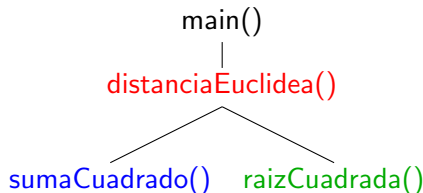
dibujo

## Ejemplo de invocación

```
tipo tpar= rec x:num; y:num;
proc distanciaEuclidea(p1:tpar, p2:tpar, var res
:num)
  a: num; b: num;
  proc sumacuadrado(a:num, b:num, var r:num)
    a:=a*a;
    b:=b*b
    r:=a+b;
  proc raizcuadrada(var n:num)
    ...
  &
  a:=p1.x-p2.x;
  b:=p1.y-p2.y;
  sumacuadrado(a, b, res);
  raizcuadrada(res);
par1:tpar; par2:tpar; resultado:num;
&
par1.x:=1; par1.y:=5;
par2.x:=8; par2.y:=12;
distanciaEuclidea(par1,par2,resultado);
```

¿Cual es el máximo nivel de anidamiento para éste programa?





¿Cual es el máximo nivel de anidamiento para éste programa? 2

# Esquema de la traducción

Dibujo de cómo queda la traducción de subprogramas y el main, y el salto con ir-a.

# Manejo de la activación y desactivación I

dibujo con la explicación de ir-ind

# Manejo de la activación y desactivación II

orden de la ejecución del código generado