Lenguajes y Compiladores

Binding

Registros de activación

Cadena dinámica

Cadena estática

UNLaM

Binding

Binding o ligadura: el momento preciso en que se conoce un atributo de un elemento de un lenguaje

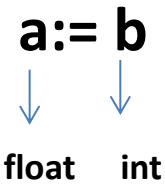
Elemento: Variables, Funciones, Parámetros, Constantes

Atributos: Tipo, Alcance, Almacenamiento, Valor

Binding de variable: el momento preciso en que se conoce el tipo de una variable de un lenguaje.

Binding de variables

Binding de Tipo



Transformar “b” para que sea float → Tipos Estáticos

Modificar el lado izquierdo para que sea entero → Tipos Dinámicos

Binding de tipo

- Una propiedad es **estática** cuando no puede cambiar durante la ejecución (no hay forma que el programador logre que cambie)

- Una propiedad es **dinámica** si puede cambiar durante la ejecución (cambia por si solo)

Binding de variables

Binding de Valor

Variables → Valor varía

Valor de las variables → Dinámico
Variable con valor estático → Constantes simbólicas

Binding de variables

Binding de Alcance

Alcance : conjunto de instrucciones en donde es posible utilizar una variable.

En lenguaje C o en Java : variables que están entre { }
variables que están dentro de una función o método

En GWBasic **:** depende

Binding de variables

Binding de Alcance

Alcance : conjunto de instrucciones en donde es posible utilizar una variable.

```
10    A=10
20    Input X
30    If X>0 then GOTO 50
40    B=7
50    C=A+B
```

¿Qué instrucciones pueden usar B ? **Depende del valor de X**

No existe una regla de alcance dinámico única, **existen varias**

Binding de variables

Binding de Alcance

Si el lenguaje posee **alcance estático** -> **no cambia** durante la ejecución siempre es posible utilizar la variable o no

Si el lenguaje posee **alcance dinámico** -> **cambia** en ***tiempo de ejecución.***

Binding de variables

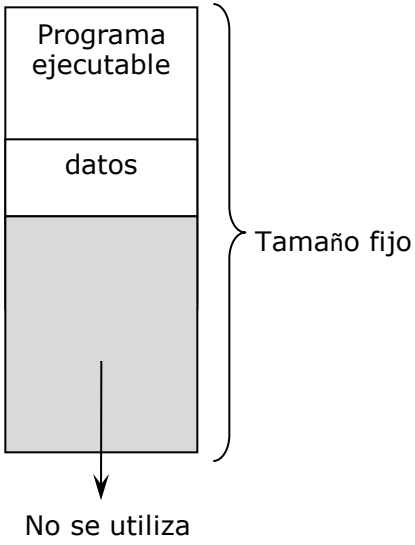
Binding de Almacenamiento

¿Cuándo una variable tiene memoria?
Las variables en C no existen hasta que una función está presente en memoria.
Las variables de Cobol están presentes siempre.

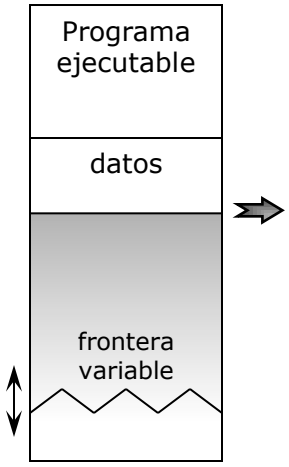
En un lenguaje con ***almacenamiento estático*** las variables tienen una **dirección fija**
En un lenguaje con ***almacenamiento dinámico***, las variables tienen una **dirección relativa**

Binding

Binding de Almacenamiento



ALMACENAMIENTO ESTÁTICO



- No se considera la memoria obtenida con malloc, new o dispose.
- No ocupan tamaño definido o fijo.
- Durante la ejecución de un programa puede ser que no entre en memoria.

ALMACENAMIENTO DINÁMICO ALCANCE y TIPO ESTÁTICO

Binding de variables

Binding de Almacenamiento

¿Cuál es el sentido el segundo dibujo?

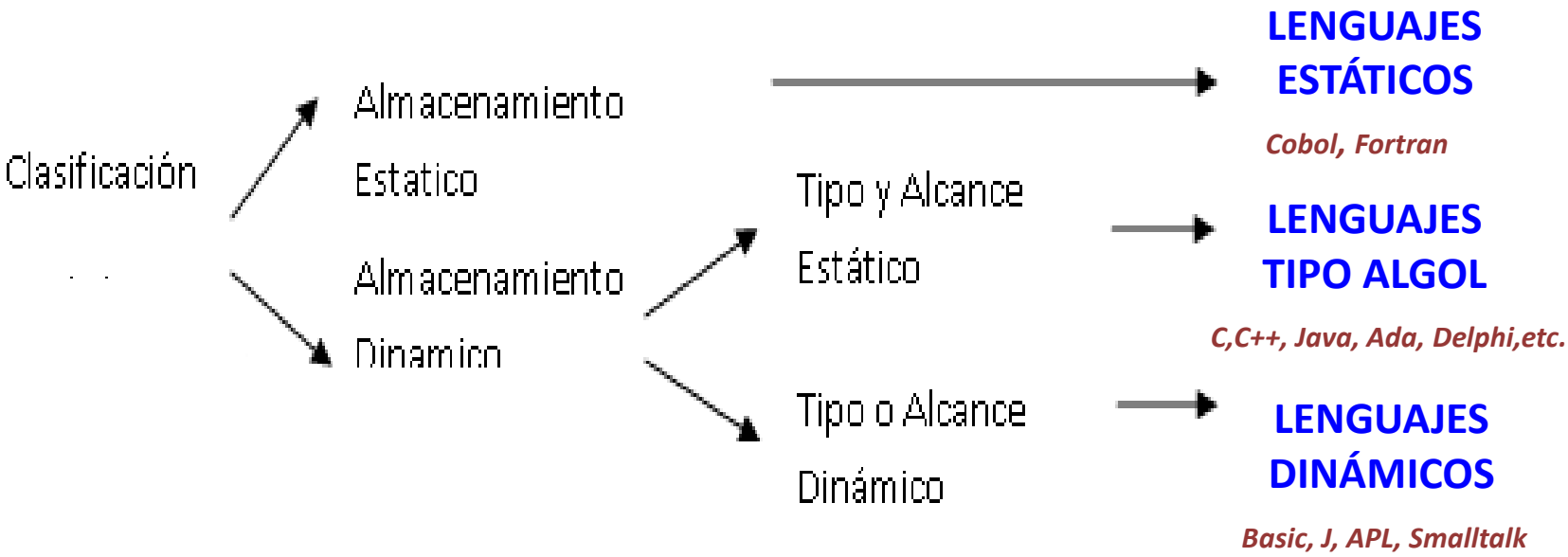
Cuando se necesita memoria, se utiliza(a demanda)

Recursividad : varios juegos de variables para una misma función

Binding de variables

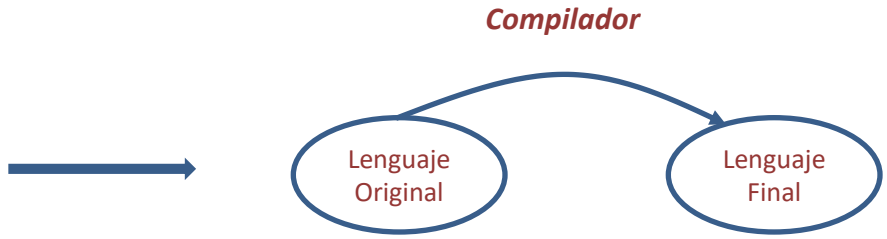
Binding de Variables	Estático	Dinámico	
Alcance	Mayoría de los lenguajes	APL/BASIC	El alcance es estático en la mayoría, y no se necesita la ejecución para saber si una variable puede invocarse en un determinado ámbito.
Valor	Constantes simbólicas	Mayoría de los lenguajes	El valor se conoce generalmente en ejecución salvo en las constantes, por ej. en C: *const int z*
Tipo	Mayoría de los lenguajes	Apl, Lisp	Lenguajes puramente dinámicos, ya que cambian su tipo en ejecución.
Almacenamiento (Memoria)	Fortran, Cobol Original	Mayoría de los lenguajes	Los que tienen binding estático de memoria tienen fijas sus direcciones de variables. La alocación se conoce en compilación. No existe la recursividad.

Clasificación de lenguajes

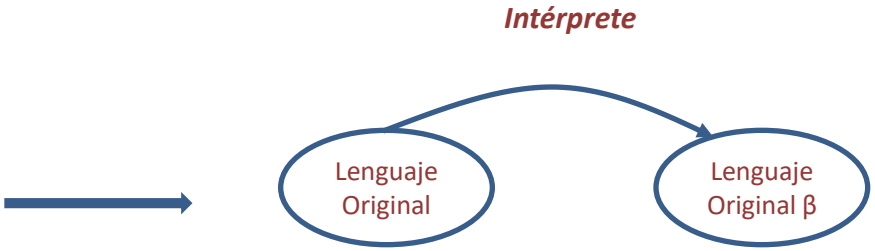


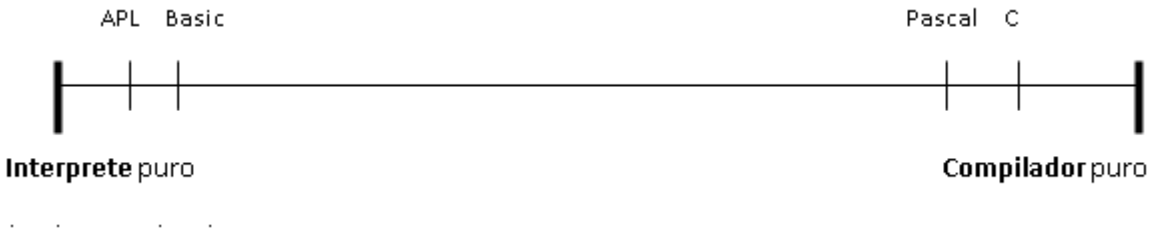
Modelos de ejecución

Modelo Compilado

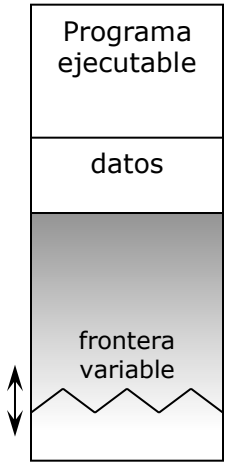


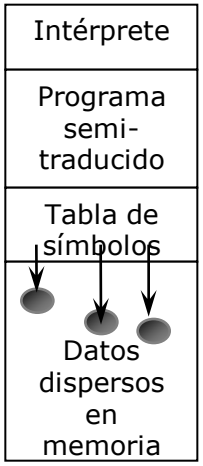
Modelo Interpretado





Tipos de lenguajes





LENGUAJES ESTÁTICOS

Almacenamiento Estático

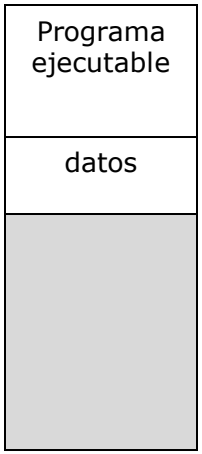
LENGUAJES ORIENTADOS A PILA (TIPO ALGOL)

Almacenamiento dinámico
Alcance y tipo estático

LENGUAJES DINÁMICOS

Almacenamiento dinámico
Alcance o tipo dinámico

Binding – Modelo de ejecución



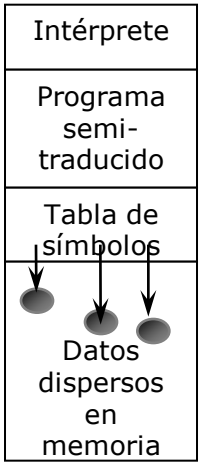
A : = B + C *(variables enteras de 16 bits)*

A ⇒ *312A*
B ⇒ *204F*
C ⇒ *2BC1*
MOV Ax , *204F*
ADD Ax , *2BC1*
MOV *312A*, Ax

LENGUAJES ESTÁTICOS

Almacenamiento Estático

Binding – Modelo de ejecución



LENGUAJES DINÁMICOS

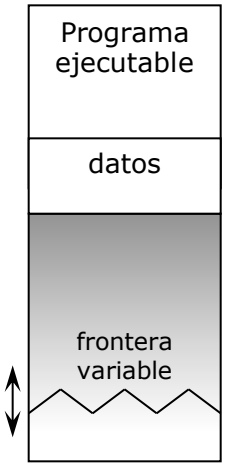
Almacenamiento dinámico
Alcance o tipo dinámico

A : = B + C *(variables enteras de 16 bits)*

El intérprete:

- Recorre la tabla de símbolos para ubicar *B*
- Recorre la tabla de símbolos para ubicar *C*
- Obtiene los tipos de datos de cada variable (enteros) y llama a la rutina sumar_enteros
- Recorre la tabla de símbolos para ubicar *A*
- Almacena el resultado de la suma en la dirección correspondiente.

Binding – Modelo de ejecución



?

LENGUAJES ORIENTADOS A PILA (TIPO ALGOL)

Almacenamiento dinámico
Alcance y tipo estático

Bloques - Unidades

Unidad: conjunto de sentencias con delimitadores de comienzo y fin en el que está permitido declarar variables locales.

Lenguaje C

```
{        -----
         -----
}
```

Lenguaje Pascal

```
FUNCTION dolares(pesos,tc:real):real;
   begin
         dolares := pesos /tc
    end;
```

Lenguaje Pascal

```
   begin
         pesos:= dolares * tc
    end;
```

(no permite unidades anónimas)

Bloques - Unidades

Anónimas	**sólo se ejecutan si el flujo del programa cae dentro de la unidad** ***Ej. { …..} en C (unidades en C)***
Con nombre	**sólo se ejecutan si se las invoca** ***Ej. Call …. o envío de mensaje***

Unidades Anónimas : no permiten recursividad

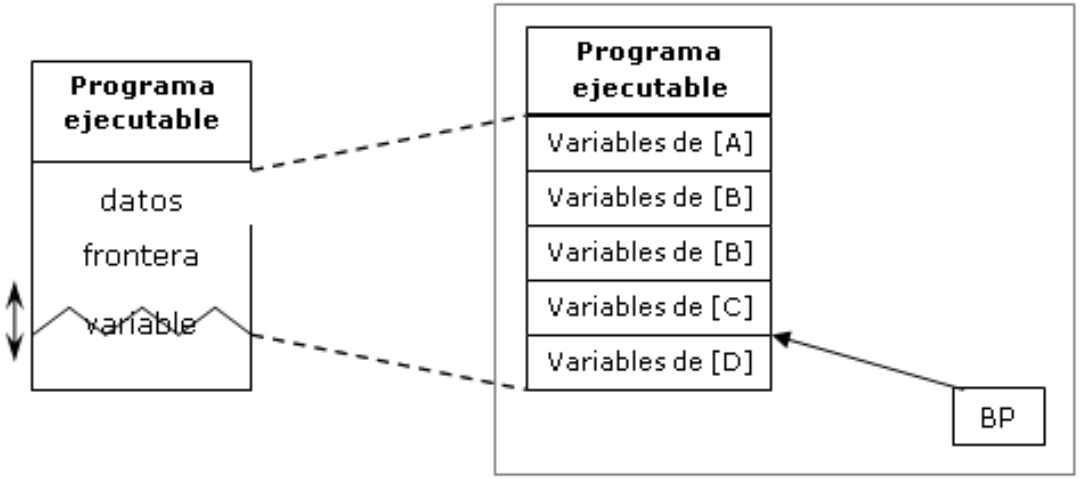
Unidad con Nombre: permiten recursividad

Bloques - Unidades

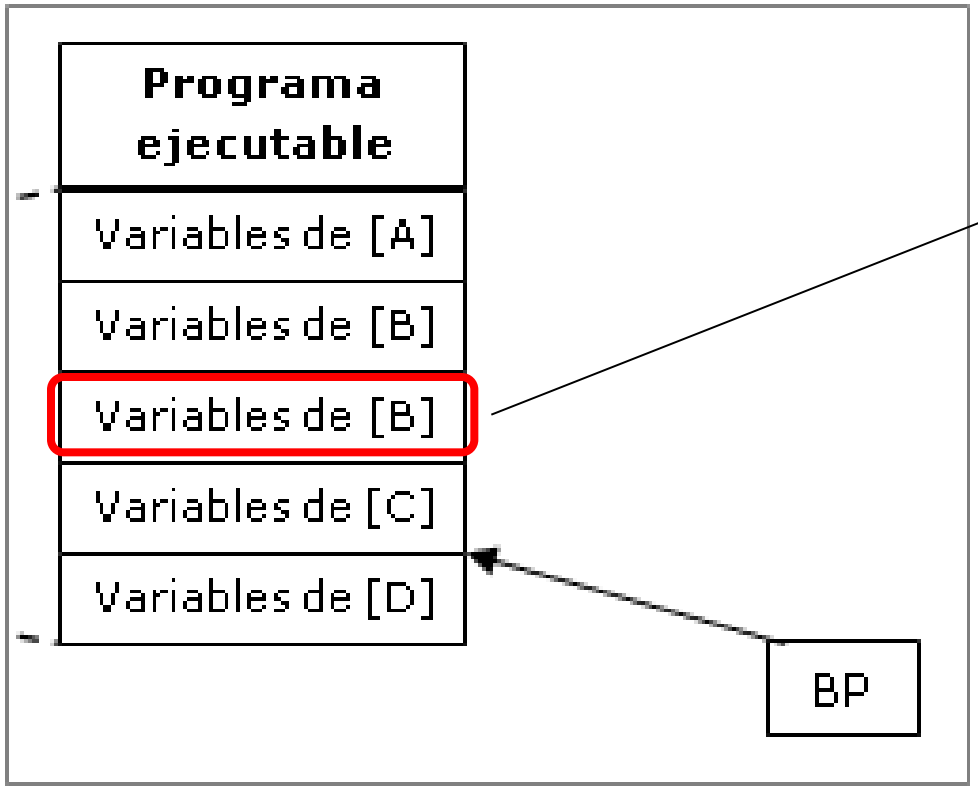
Supóngase que ***durante la ejecución***, existen distintas unidades que se llaman ($\rightarrow$) unas a otras

$$A \rightarrow B \rightarrow B \rightarrow C \rightarrow D$$

Durante la ejecución de una unidad, adquieren memoria las variables locales de la misma.



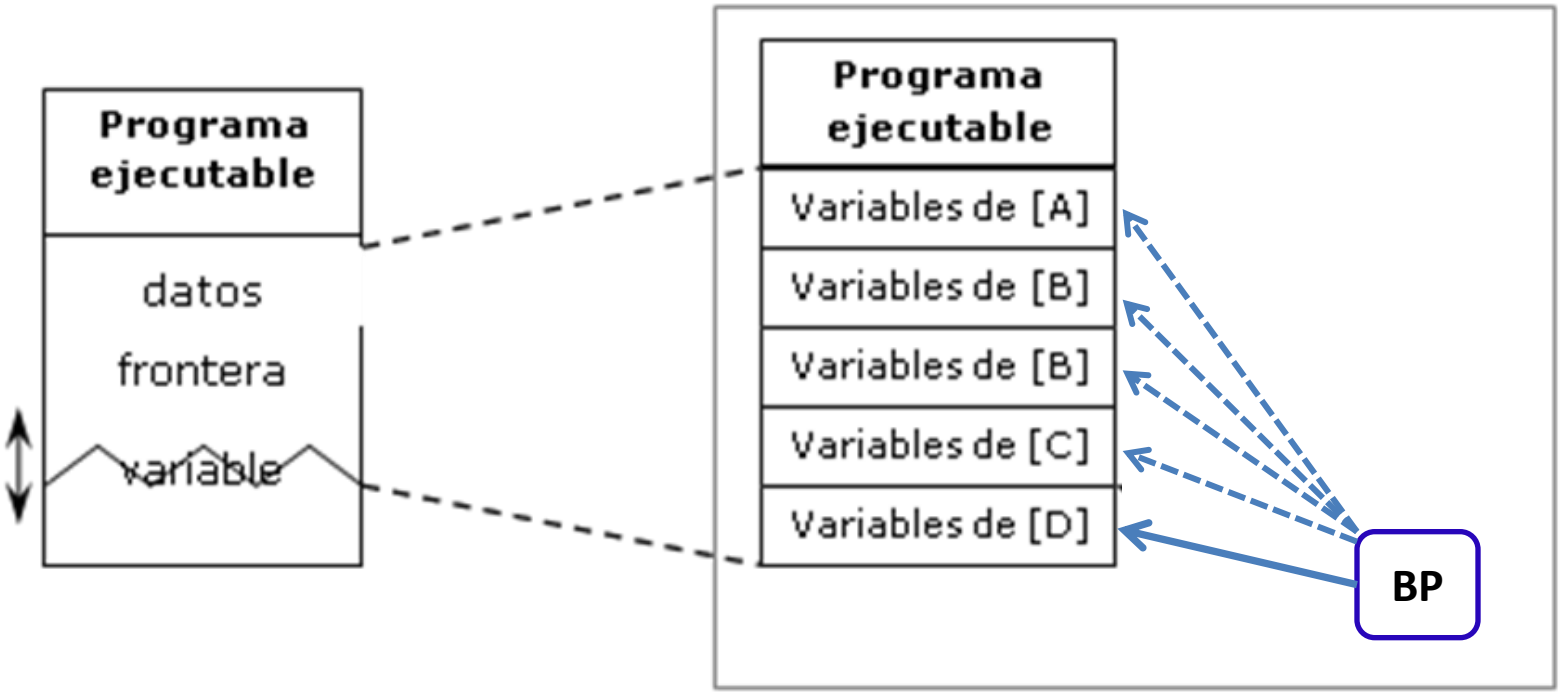
Bloques - Unidades



Registro de Activación

Conjunto de datos nuevos necesarios para comenzar la ejecución de una unidad

Registro de activación

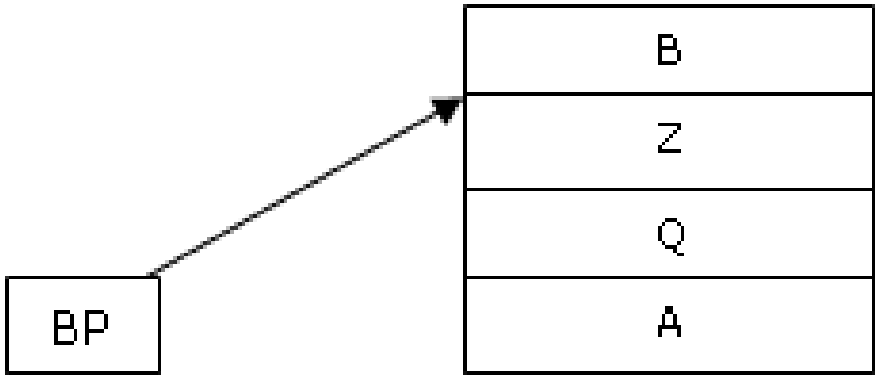


BP (Base Pointer) → *variable estática fija en dirección cuyo contenido es la dirección base del Registro de Activación*

Registro de activación

A→ Q→ Z → B

b, c y ***a*** declaradas en B
a offset 22
b offset 44
c offset 60
(desde el comienzo del RA)



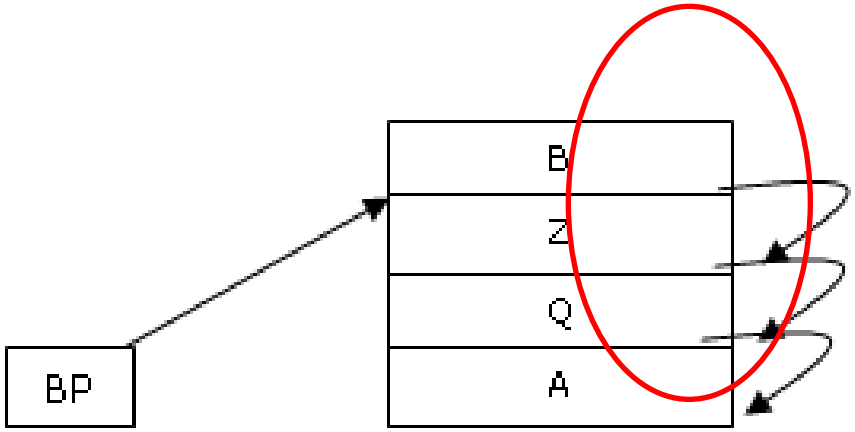
Instrucción → ***a := b + c*** en B

MOV R1, BP+44
ADD R1, BP+60
MOV BP+22, R1

DIRECCIONAMIENTO RELATIVO O INDEXADO

¿Cómo se traducen las direcciones de estas variables (locales) ?
Offset : distancia desde la base del RA de la unidad donde fueron declaradas

Cadenas

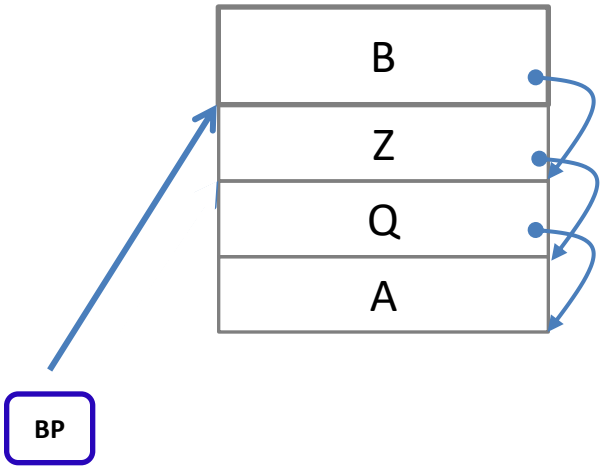


CADENA DINÁMICA

Lugar fijo (en distancia) en cada registro de activación que guarda la dirección de la unidad anterior

Offset Cadena Dinámica : 10

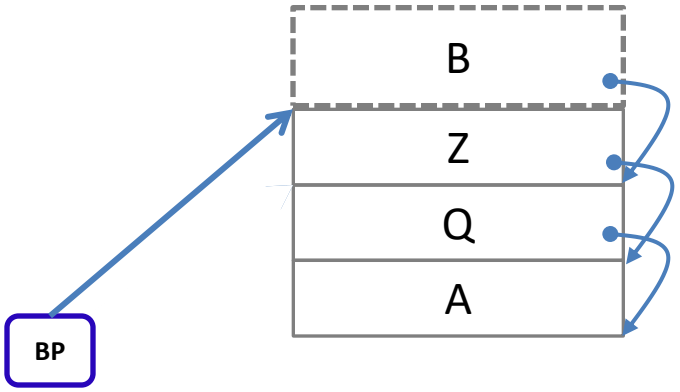
Z → B



¿Qué tuvo que haber agregado el compilador al ejecutable?

MOV R1, BP
ADD BP, tamaño(Z)
MOV BP+10, R1

Cadenas

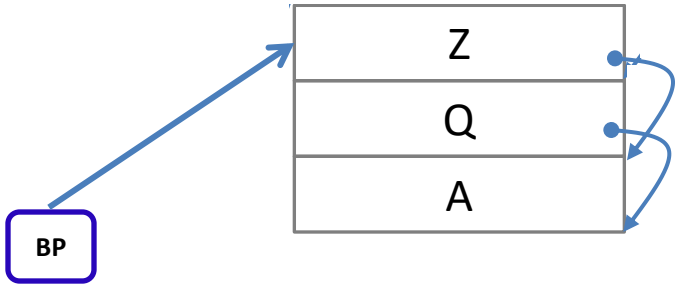


Z → B *(hubo un call B desde Z en el pgm. fuente) …*

MOV R1, BP *(Resguarda el BP)*

ADD BP, tamaño(Z) *(Crea el RA (B))*

MOV BP+10, R1 *(Crea la CD en B)*

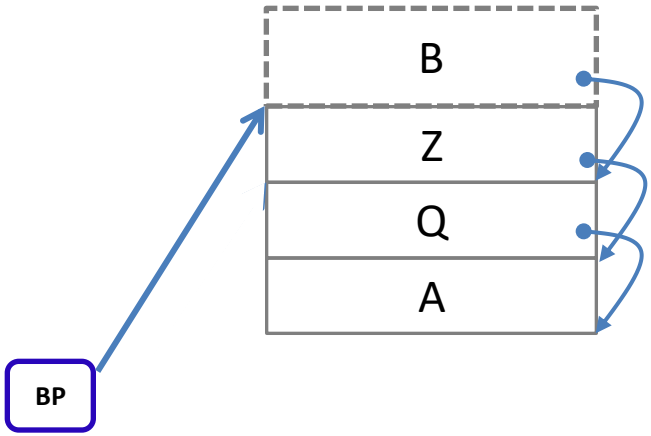


Cuándo B termina ...

MOV BP, BP+10

Tipos de variables

Sintetizando:



Programa Fuente

```
function Z ()
        call B ()
        -
        -
function B ()
        int a,b,c
        a := b + c
return
```

Programa Compilado

```
MOV R1, BP
ADD BP, tamaño(Z)
MOV BP+10, R1
```
Crea RA (B)

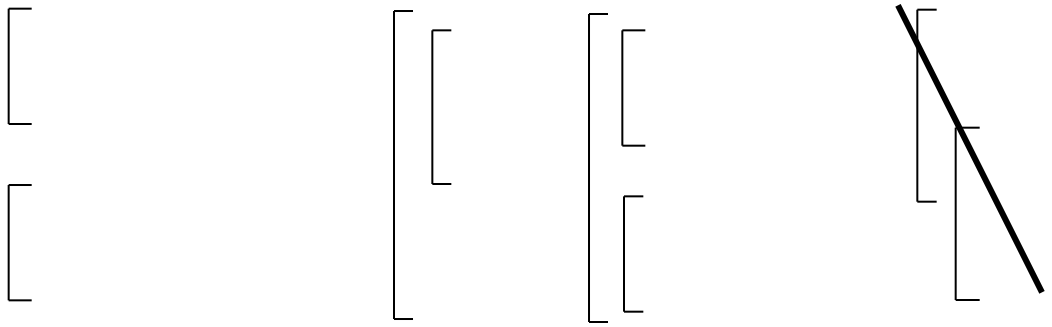
Crea CD (B)

```
MOV R1, BP+22
ADD R1, BP+44
MOV BP+60, R1
```

```
MOV BP, BP+10
```
Restaura a RA(Z)

Cadenas

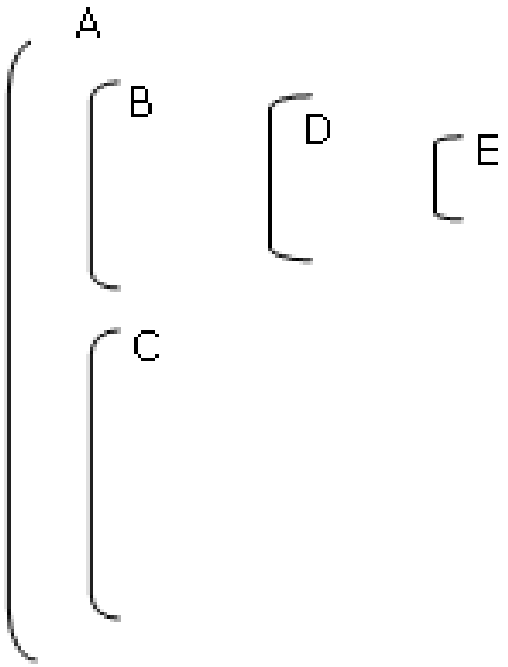
Volvamos a las unidades…..

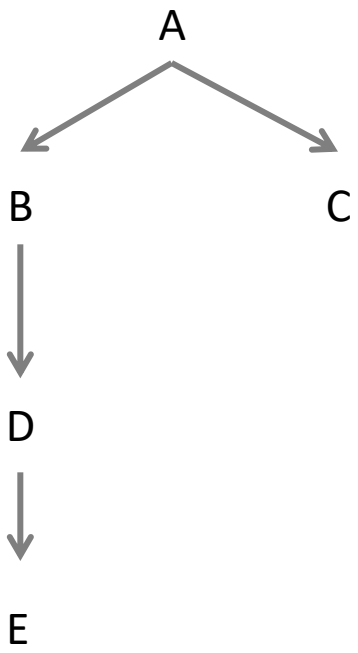


Las unidades dentro de un programa, son totalmente separadas o una unidad está completamente dentro de otra, es decir, son anidadas o disjuntas (no comparten nada)

Cadenas

Supongamos el siguiente esquema de un programa





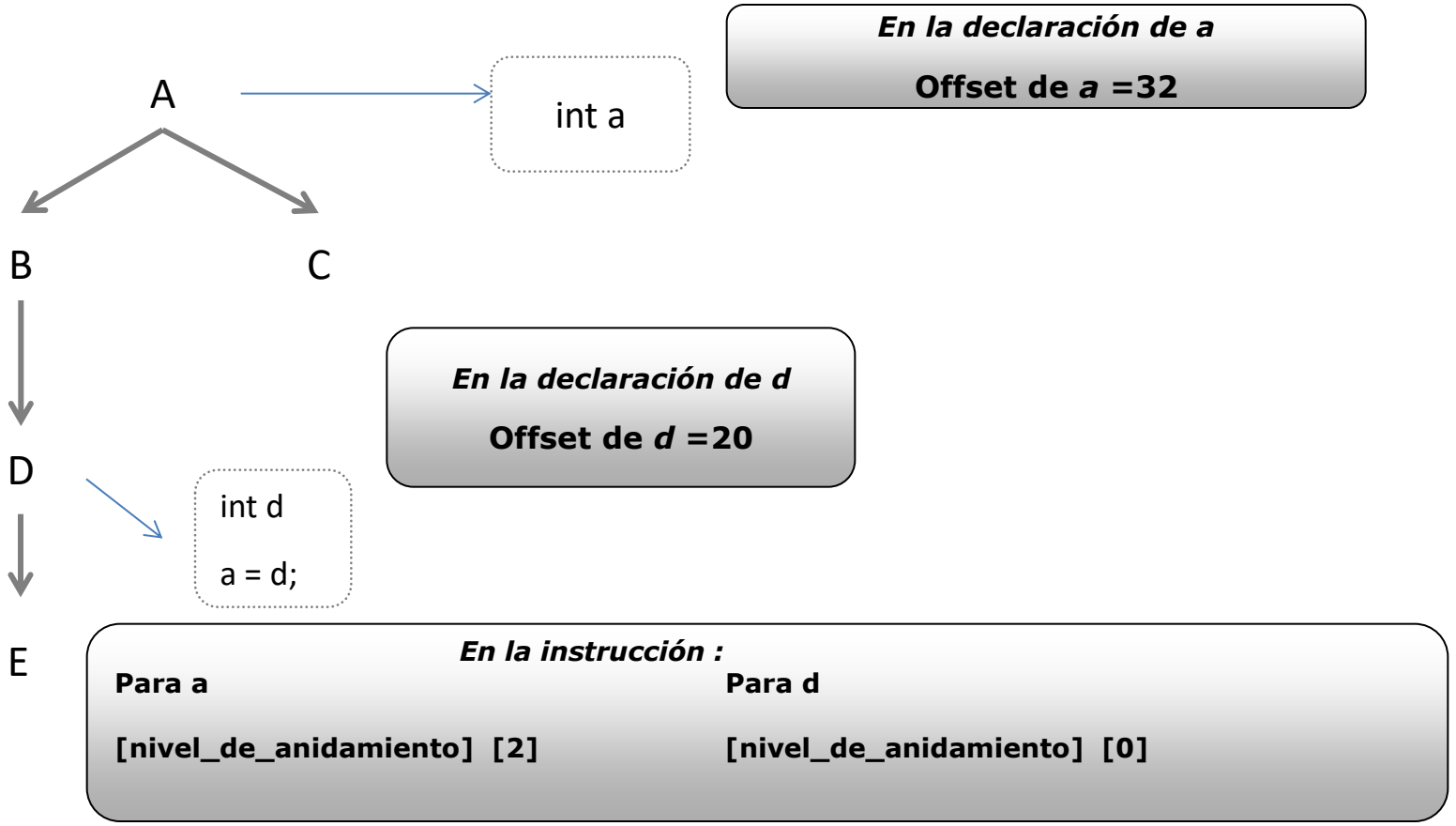
Las reglas de alcance estático, en general, establecen que todo nombre que no es encontrado en el bloque de su declaración, deberá ser buscado en el bloque inmediato en el que se encuentra anidado, hasta el más externo.

Cadenas

¿Cómo se traducen las direcciones de las variables?

Offset : distancia desde la base del RA de la unidad donde fueron declaradas

Anidamiento: cantidad de niveles de anidamiento desde la unidad donde es utilizada la variable hasta la unidad donde se encuentra declarada



Cadenas

La instrucción citada anteriormente…(slide 24)

A→ Q→ Z → B
Instrucción → ***a := b + c*** en B

MOV R1, BP+44
ADD R1, BP+60
MOV BP+22, R1

Es correcta si se trata de variables locales, pero …

Si ***a*** o ***b*** o ***c*** no están en B

El código no funciona

Cadenas

Cadena Estática

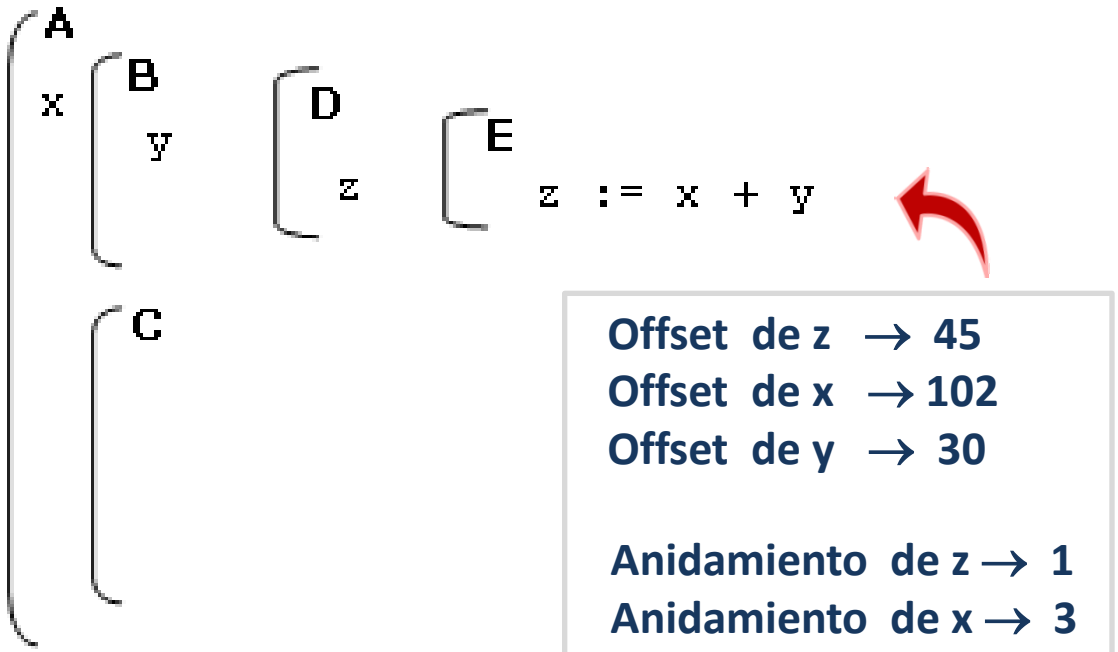
Teniendo en cuenta la regla de alcance, el acceso a las variables globales puede ser implementado mediante el mecanismo de cadena estática

La cadena estática es una cadena de enlaces que conecta distintas instancias de registros de activación en la pila.
Con un lugar específico en el registro de activación de cada bloque, cada puntero de esta cadena apunta específicamente a la base del registro de activación de su bloque padre.

El nivel de anidamiento de una unidad y todas las variables utilizadas dentro de él, es la cantidad de enlaces a seguir dentro de la cadena estática

Cadenas

Secuencia de llamadas A → C → B → D → E



Offset de z → 45
Offset de x → 102
Offset de y → 30

Anidamiento de z → 1
Anidamiento de x → 3
Anidamiento de y → 2

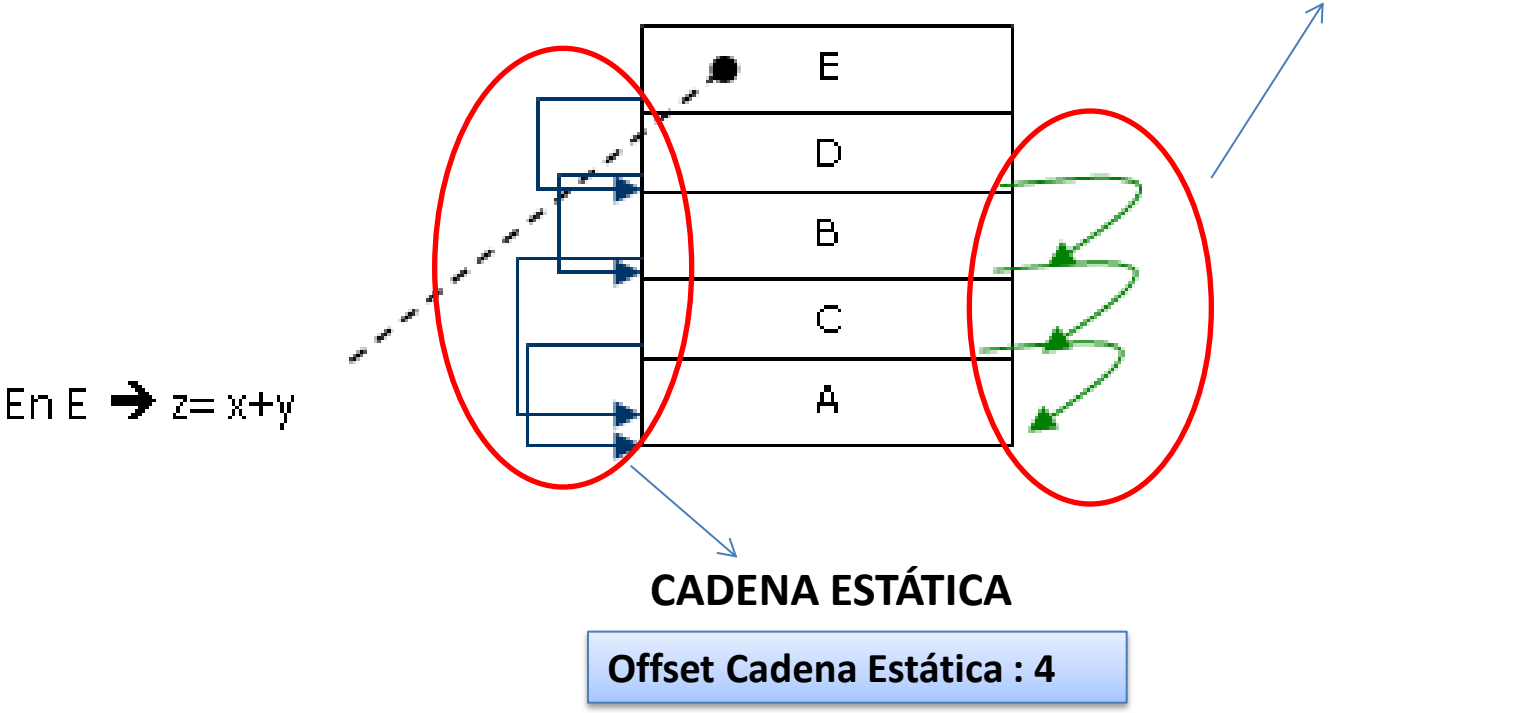
El anidamiento cambiaría si la instrucción no estuviese en E

Cadenas

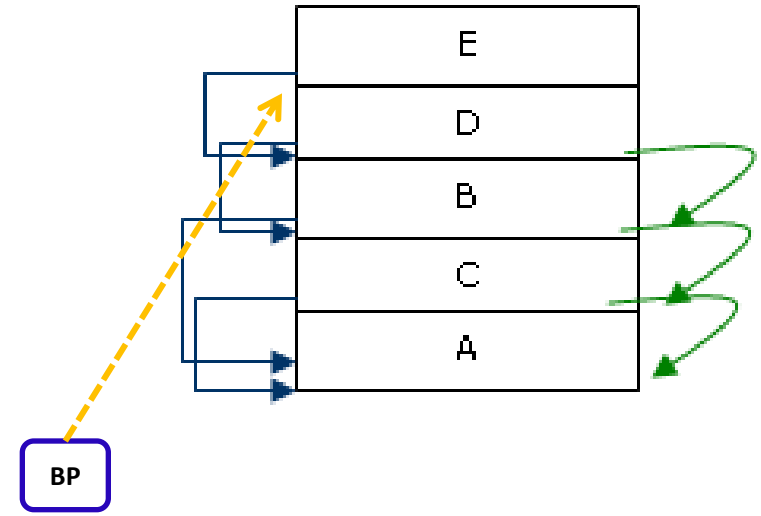
Veamos la situación cuando E se está ejecutando…

CADENA DINÁMICA

Offset Cadena Dinámica : 10



Cadenas



A
x
B
y
D
z
E
z := x + y
C

Offset de z → 45
Offset de x → 102
Offset de y → 30

Anidamiento de z → 1
Anidamiento de x → 3
Anidamiento de y → 2

MOV R2, BP	Resguardo de BP (base de E)
MOV BP, BP+4 MOV BP, BP+4 MOV BP, BP+4	3 Saltos por cadena estática, distancia 4 (llega a base de A)
MOV R1, BP + 102	Almacena x en R1 (offset de x= 102)
MOV BP, R2	Retorna el BP a la base de E
MOV BP, BP+4 MOV BP, BP+4	2 Saltos por cadena estática, distancia 4 (llega a base de B)
ADD R1, BP + 30	Suma a R1 el valor de y (offset de y=30)
MOV BP, R2	Retorna el BP a la base de E
MOV BP, BP+4	1 Salto por cadena estática, distancia 4 (llega a base de D)
MOV BP + 45, R1	Almacena en z la suma de x+y (offset de z=45)
MOV BP, R2	Retorna el BP a la base de E

Cadenas

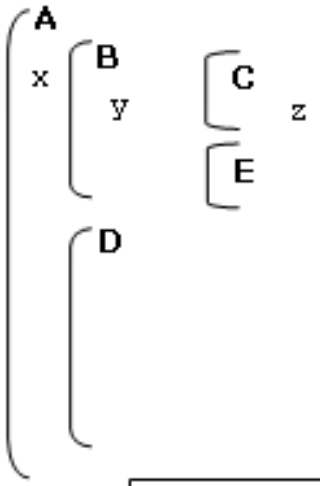
Pero…

¿Cómo se construyen los punteros que forman la cadena estática?

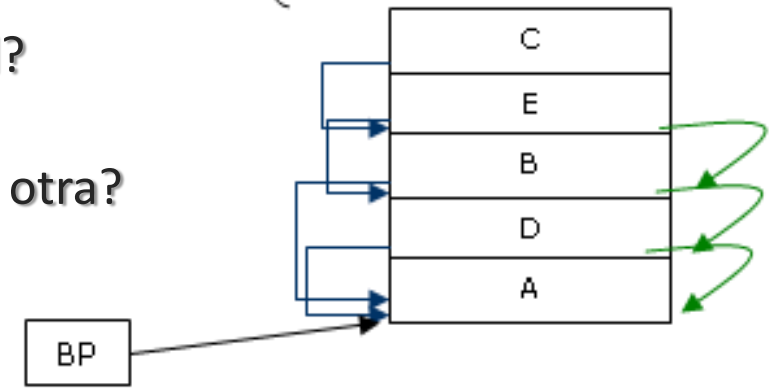
Cadena estática

Supongamos la siguiente secuencia de llamados A → D → B → E → C dónde :

***x* está declarada en A**
***y* en B**
***z* en C**

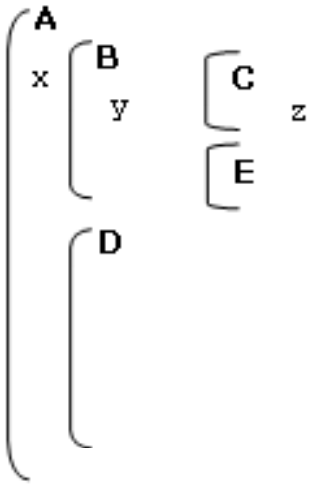


- ¿Cuál es el alcance de una unidad?
- ¿Puede cualquier unidad llamar a otra?



Cadena estática

La respuesta es… no siempre



→	A	B	C	D	E
A	R	L	X	L	X
B	G_2	R	L	G_1	L
C	G_3	G_2	R	G_2	G_1
D	G_2	G_1	X	R	X
E	G_3	G_2	G_1	G_2	R

Cadena estática

Construcción

Secuencia de llamados A → D → B → E → C y C → D

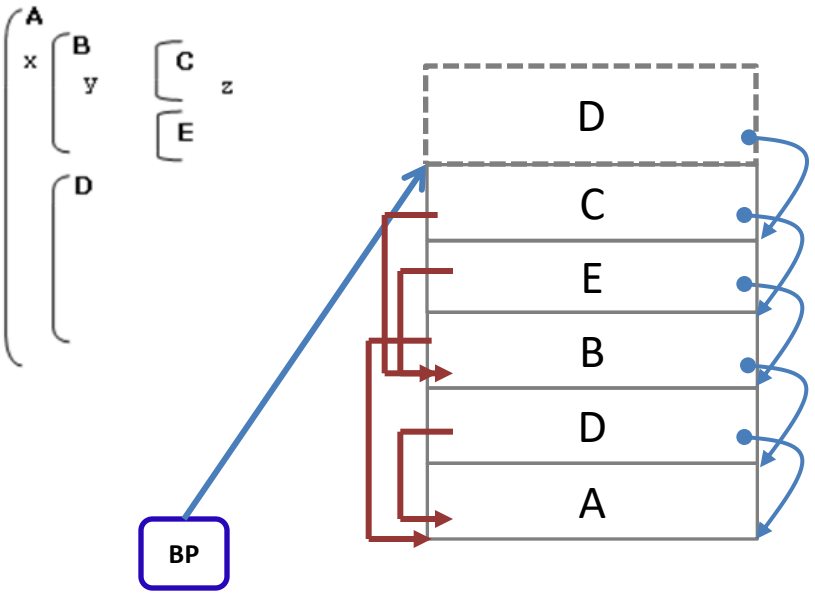
(Luego de la construcción de la CD…)

Offset Cadena Dinámica : 10

MOV R1, BP

Resguarda el BP (base de D)

Offset Cadena Estática : 4



Cadena estática

Construcción

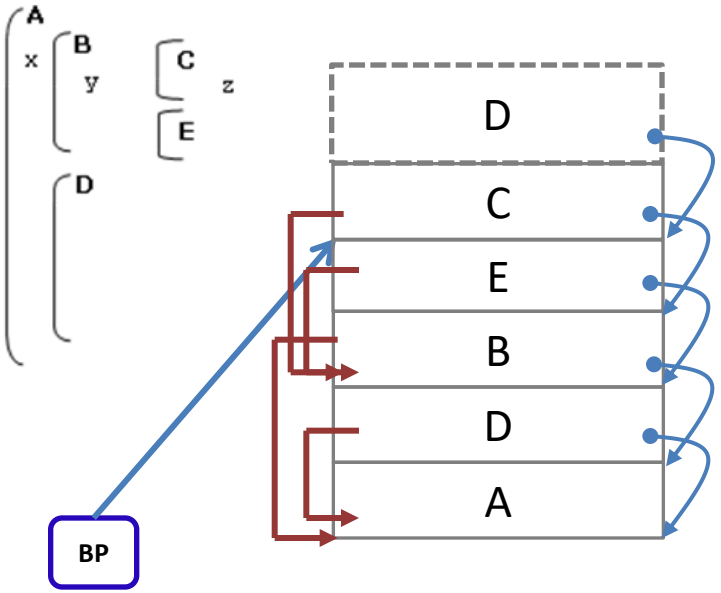
Secuencia de llamados A → D → B → E → C y C → D

Offset Cadena Dinámica : 10

Offset Cadena Estática : 4

MOV R1, BP — Resguarda el BP (base de D)

MOV BP, BP+10 — Copiar CD en BP (base de C)



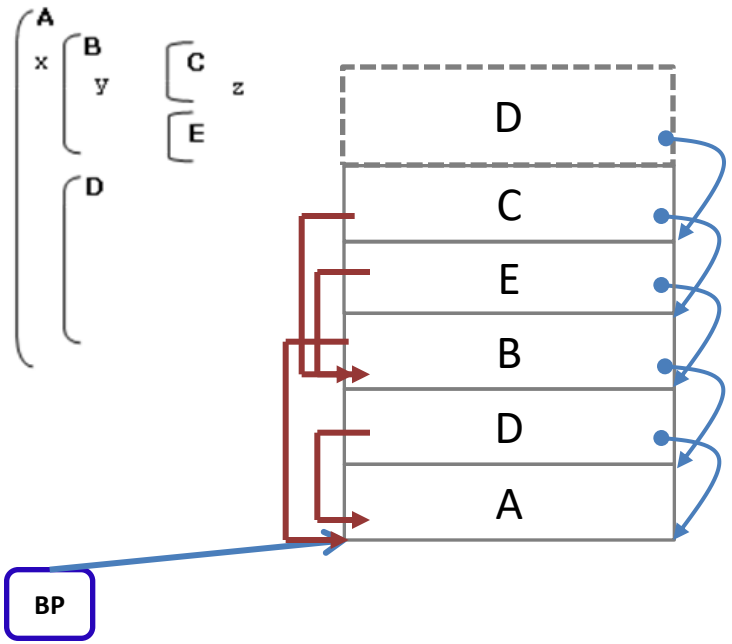
Cadena estática

Construcción

Secuencia de llamados A → D → B → E → C y C → D

Offset Cadena Dinámica : 10

Offset Cadena Estática : 4



MOV R1, BP	Resguarda el BP (base de D)
MOV BP, BP+10	Copiar CD en BP (base de C)
MOV BP, BP+4 MOV BP, BP+4	C → D es G_2 por lo tanto 2 veces la instrucción. Se llega a la base de A

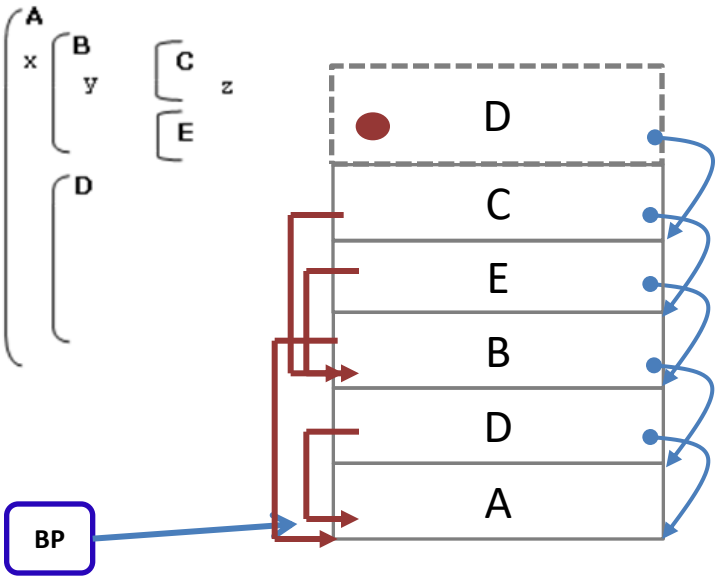
Cadena estática

Construcción

Secuencia de llamados A → D → B → E → C y C → D

Offset Cadena Dinámica : 10

Offset Cadena Estática : 4



MOV R1, BP	Resguarda el BP (base de D)
MOV BP, BP+10	Copiar CD en BP (base de C)
MOV BP, BP+4 MOV BP, BP+4	C → D es G_2 por lo tanto 2 veces la instrucción. Se llega a la base de A
MOV R1+4,BP	Copia la dirección base de A en la cadena estática de D

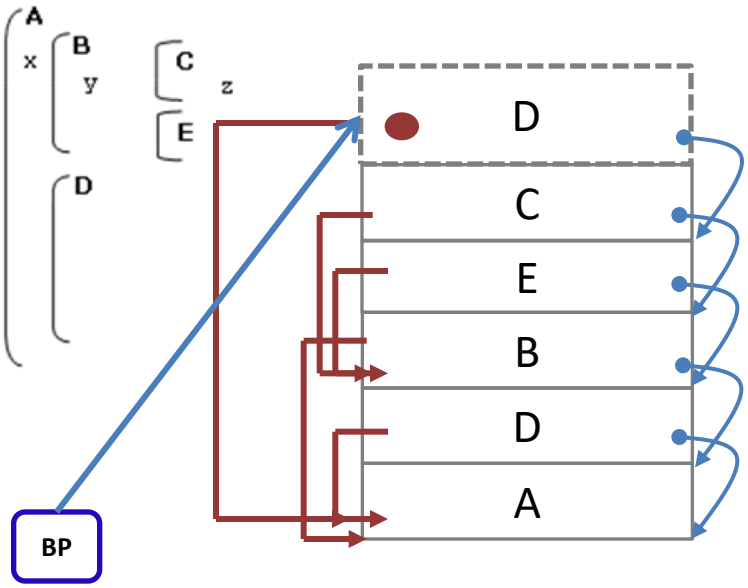
Cadena estática

Construcción

Secuencia de llamados A → D → B → E → C y C → D

Offset Cadena Dinámica : 10

Offset Cadena Estática : 4



MOV R1, BP	Resguarda el BP (base de D)
MOV BP, BP+10	Copiar CD en BP (base de C)
MOV BP, BP+4 MOV BP, BP+4	C → D es G_2 por lo tanto 2 veces la instrucción. Se llega a la base de A
MOV R1+4,BP	Copia la dirección base de A en la cadena estática de D
MOV BP, R1	Restaura el BP a la base de D

¿Preguntas?