

CHƯƠNG 7: ĐIỀU KHIỂN DỮ LIỆU

Nội dung

- Các khái niệm cơ bản
- Dữ liệu cục bộ và môi trường cục bộ
- Dữ liệu dùng chung: Truyền thông số
- Dữ liệu dùng chung: Tầm vực động
- Dữ liệu dùng chung: Cấu trúc khối

Truyền dữ liệu

- Truyền trực tiếp
 - $3 + 4 * 7$
- Truy xuất đtdl thông qua tên

Tên

- Tên đơn giản
 - a,b
- Tên phức hợp: kết hợp với tác vụ lựa chọn và truy xuất thành phần
 - a[1]
 - b.c[2]

Kết hợp và Khai báo

- Khai báo: sự ràng buộc giữa tên và các tính chất của nó
 - Vd: `var a:integer;` ràng buộc a với biến nguyên
- Kết hợp: sự ràng buộc giữa tên và đtdl hoặc một chương trình con cụ thể
 - Vd: a

obj1

Môi trường tham khảo

- Môi trường tham khảo tĩnh: tập hợp các khai báo mà 1 chương trình có thể tham khảo
- Môi trường tham khảo động: tập hợp các kết hợp mà 1 bản hoạt động của chương trình con có thể tham khảo
- Các loại môi trường tham khảo
 - Môi trường cục bộ
 - Môi trường không cục bộ
 - Môi trường toàn cục
 - Môi trường định nghĩa trước

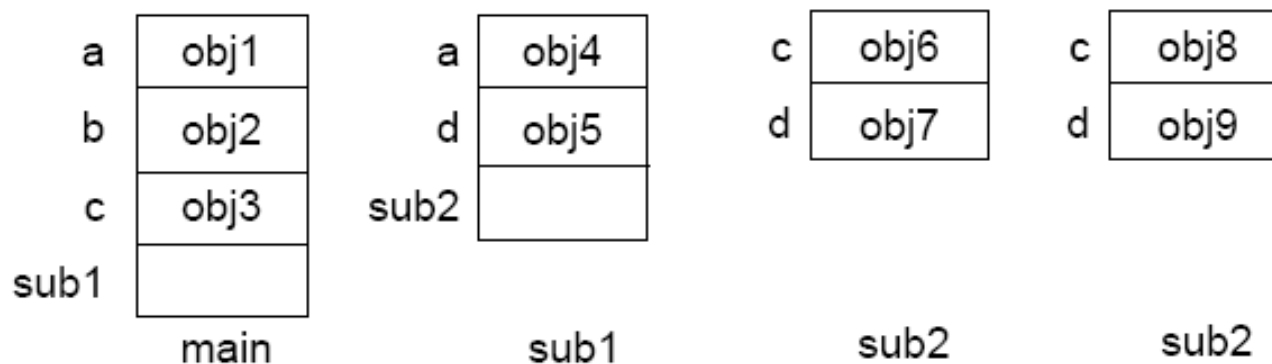
Vấn đề khi đặc tả tác vụ

- program main;
- var A, B, C: real;
- procedure Sub1 (A: real);
 - var D: real;
 - procedure Sub2 (C: real);
 - var D: real;
 - begin
 - ...C:= C+B; ...
 - end;
 - begin
 - ...Sub2(B); ...
 - end;
- begin
- ...Sub1(A); ...
- end.

	Cục bộ	Không cục bộ	Toàn cục
Sub2	C, D	A, Sub2 B, Sub1	B, Sub1
Sub1	A, D, Sub2	B, C, Sub1	B, C, Sub1
main	A, B, C, Sub1		A, B, C, Sub1

Môi trường tham khảo tĩnh

main → sub1 → sub2 → sub2



	cục bộ	không cục bộ	toàn cục
sub2	obj8,obj9	obj4, obj2,sub1,sub2	obj2,sub1
sub2	obj6,obj7	obj4, obj2,sub1,sub2	obj2,sub1
sub1	obj4, obj5,sub2	obj2,obj3,sub1	obj2, obj3,sub1
main	obj1,obj2,obj3,sub1		obj1,obj2,obj3,sub1

Môi trường tham khảo động

Khả kiến và Bị che

- Một khai báo (hoặc kết hợp) có thể tham khảo bởi 1 chương trình con: khả kiến
- Một khai báo tồn tại nhưng không khả kiến: bị che
 - Vd: kết hợp D với biến cục bộ trong Sub1 là bị che với Sub2

Tầm vực

- Tầm vực của một ràng buộc là phạm vi chương trình (định nghĩa hoặc bản hoạt động) mà ràng buộc có hiệu lực.
- Tầm vực tĩnh
- Tầm vực động

Cấu trúc khối

- Chương trình gồm nhiều khối lồng nhau
- Khối bắt đầu bằng khai báo, theo sau là các phát biểu
- Khối tạo ra môi trường cục bộ của khối
- Quy tắc tầm vực tĩnh của nn cấu trúc khối
 - Khai báo đầu khối tạo môi trường tham khảo cục bộ của khối
 - Khối bị bao thừa hưởng các khai báo của khối bao nó
 - Mọi khai báo của khối là bị che với các khối bên ngoài
 - Một khối có thể được đặt tên – tên khối thuộc môi trường cục bộ của khối bao trực tiếp khối đang xét

Ví dụ

- var a: integer
 - procedure sub(a: integer) //sub-1
 - procedure sub(a: integer) //sub-2
 - begin
 - ...
 - sub(1); //1
 - end
 - begin
 - ...
 - sub(1); //2
 - end
- begin
 - sub(1); //3
- end

Dữ liệu cục bộ và môi trường cục bộ

- Dữ liệu cục bộ: gồm các thông số hình thức, khai báo biến và chương trình con ở đầu chương trình con đang xét
- Mttk tĩnh: các khai báo cục bộ ở đầu khối
- Mttk động: các kết hợp với các đtdl hoặc ct con ở trong bản hoạt động của ct con đó
- Qui tắc tầm vực tĩnh: mọi tham khảo đến một danh hiệu X đều liên hệ với khai báo X ở đầu khối
- Qui tắc tầm vực động: mọi tham khảo đến danh hiệu X đều liên hệ đến kết hợp giữa X với đtdl hoặc ct con trong bản hoạt động tương ứng

Hiện thực

- Tĩnh: Bảng danh hiệu để lưu trữ các ràng buộc giữa danh hiệu và các thông tin tương ứng
- Động: Dựa vào địa chỉ bản hoạt động (lưu tại CEP) + độ dời

Ví dụ

```
var a,b:integer;  
    c:char;  
begin  
    a := b - 1;  
    ...  
end
```

Dịch

a	var	integer	0
b	var	integer	2
c	var	char	4

Bảng danh hiệu

Thực thi

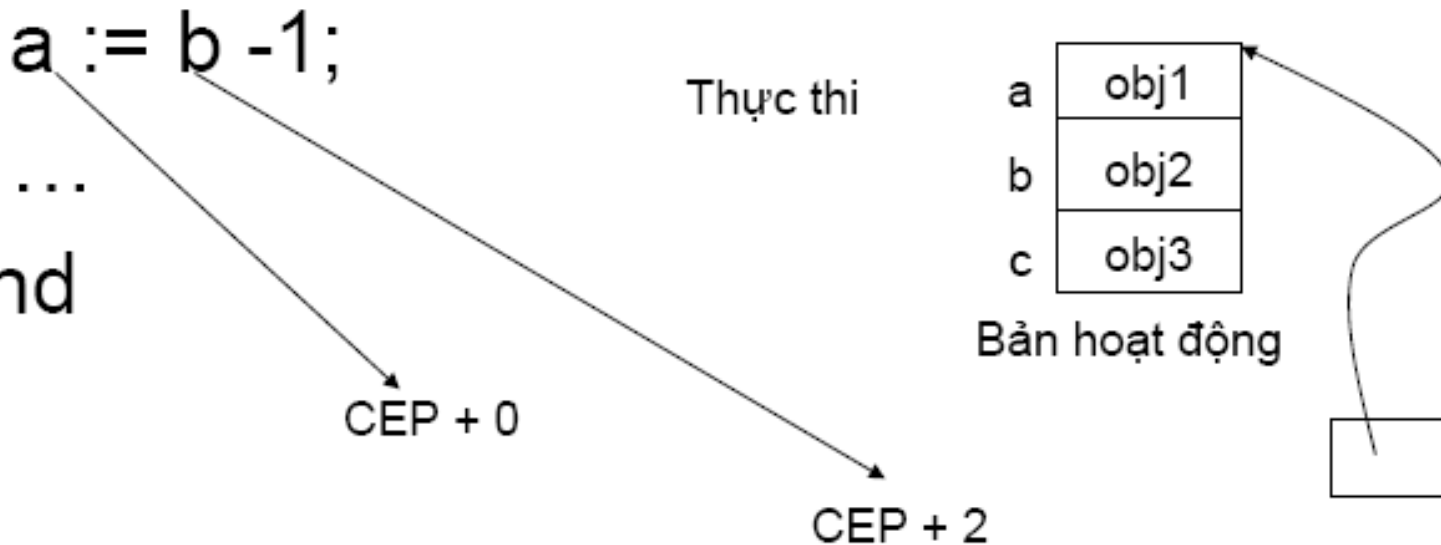
a	obj1
b	obj2
c	obj3

Bản hoạt động

CEP + 0

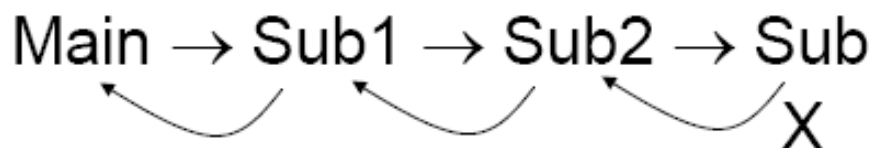
CEP + 2

CEP



Dữ liệu dùng chung: Tầm vực động

- Tham khảo không cục bộ \Rightarrow khai báo và kết hợp nào tương ứng với tham khảo này ???
- Một số ngôn ngữ (APL, LISP, SNOBOL) không có qui tắc tầm vực tĩnh mà chỉ có qui tắc tầm vực động
- Dựa vào dây xích động: đi ngược theo dây xích động, tìm kết hợp đầu tiên tương ứng với danh hiệu được tham khảo



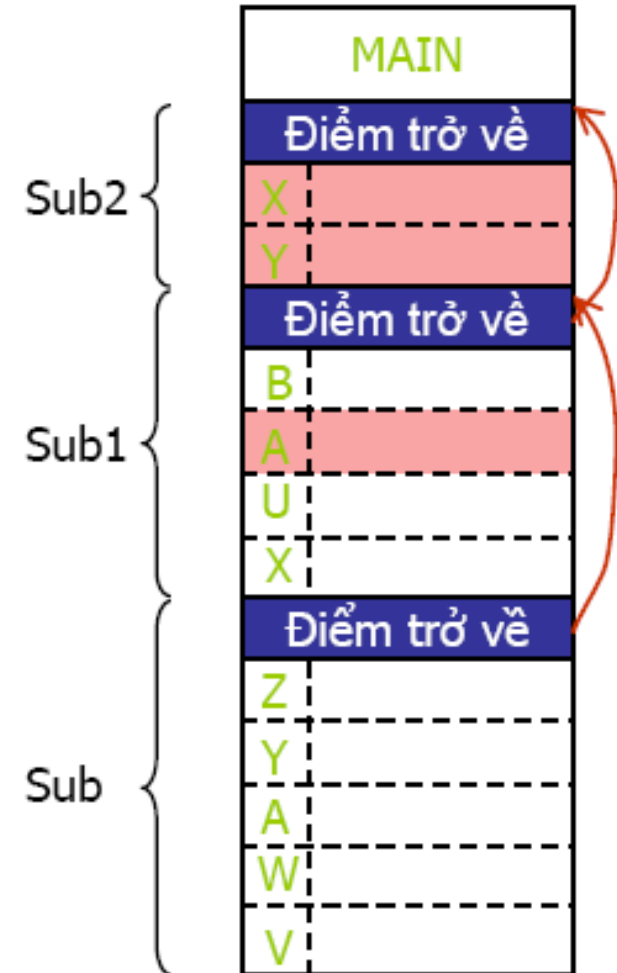
Ví dụ

```

Main() {
    ...
    Sub2()
}
Sub2 () {
    var X,Y
    ...Sub1()
}
Sub1 () {
    var B,A,U,X
    ...Sub()
}
Sub () {
    var Z,Y,A,W,V
    ...X...B...U → X,B,U của Sub1
}

```

X có thể của Sub2 ?

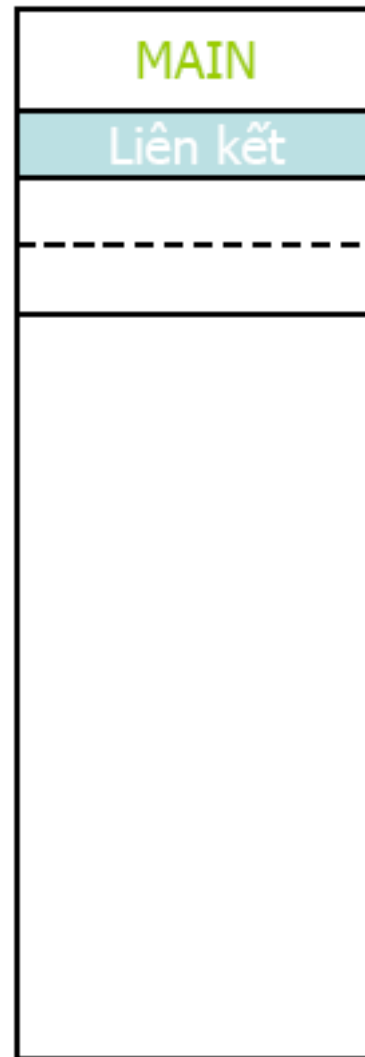


Hiện thực bằng bảng trung tâm MAIN

Chồng ấn



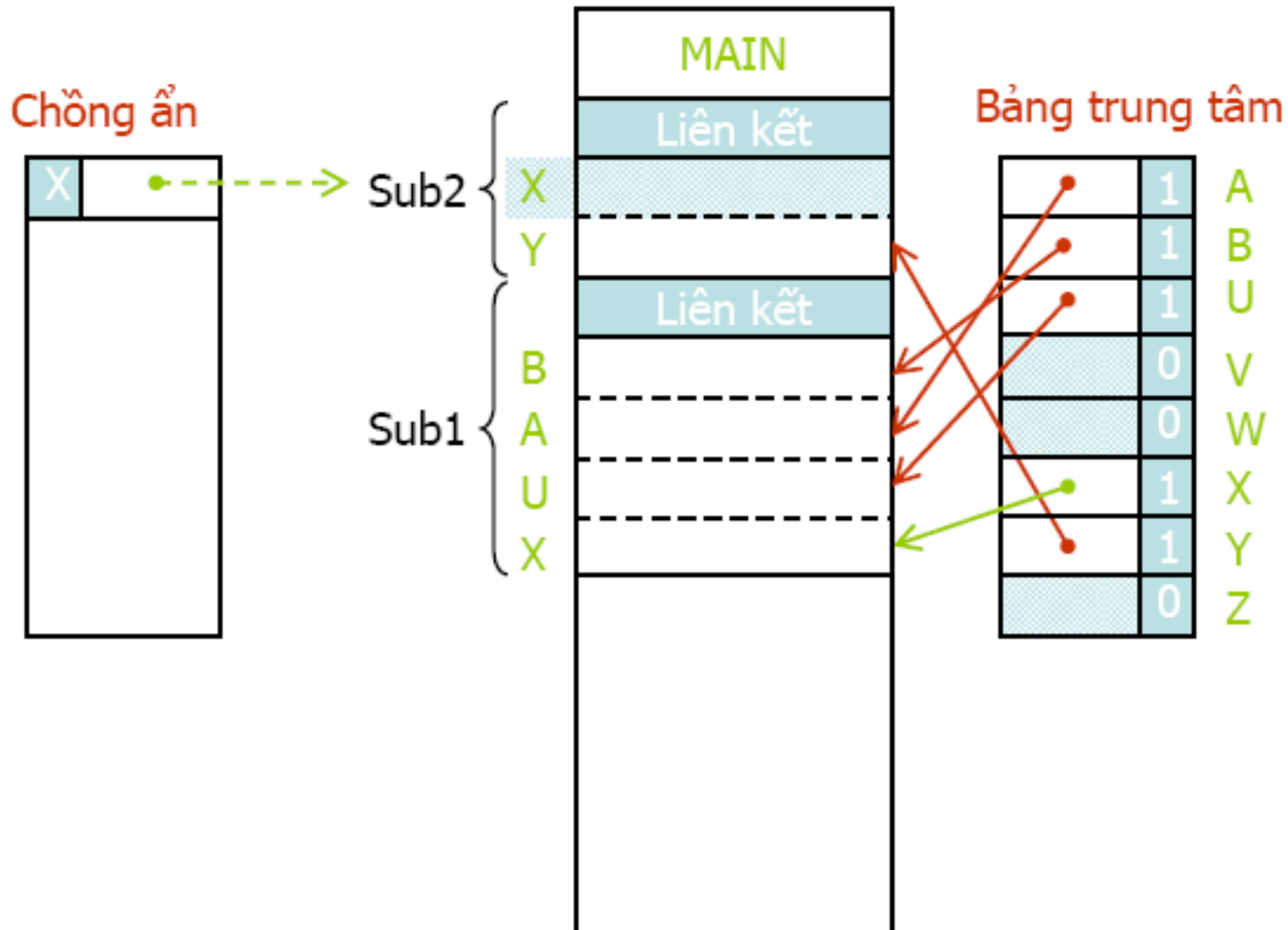
Sub2 {
X
Y



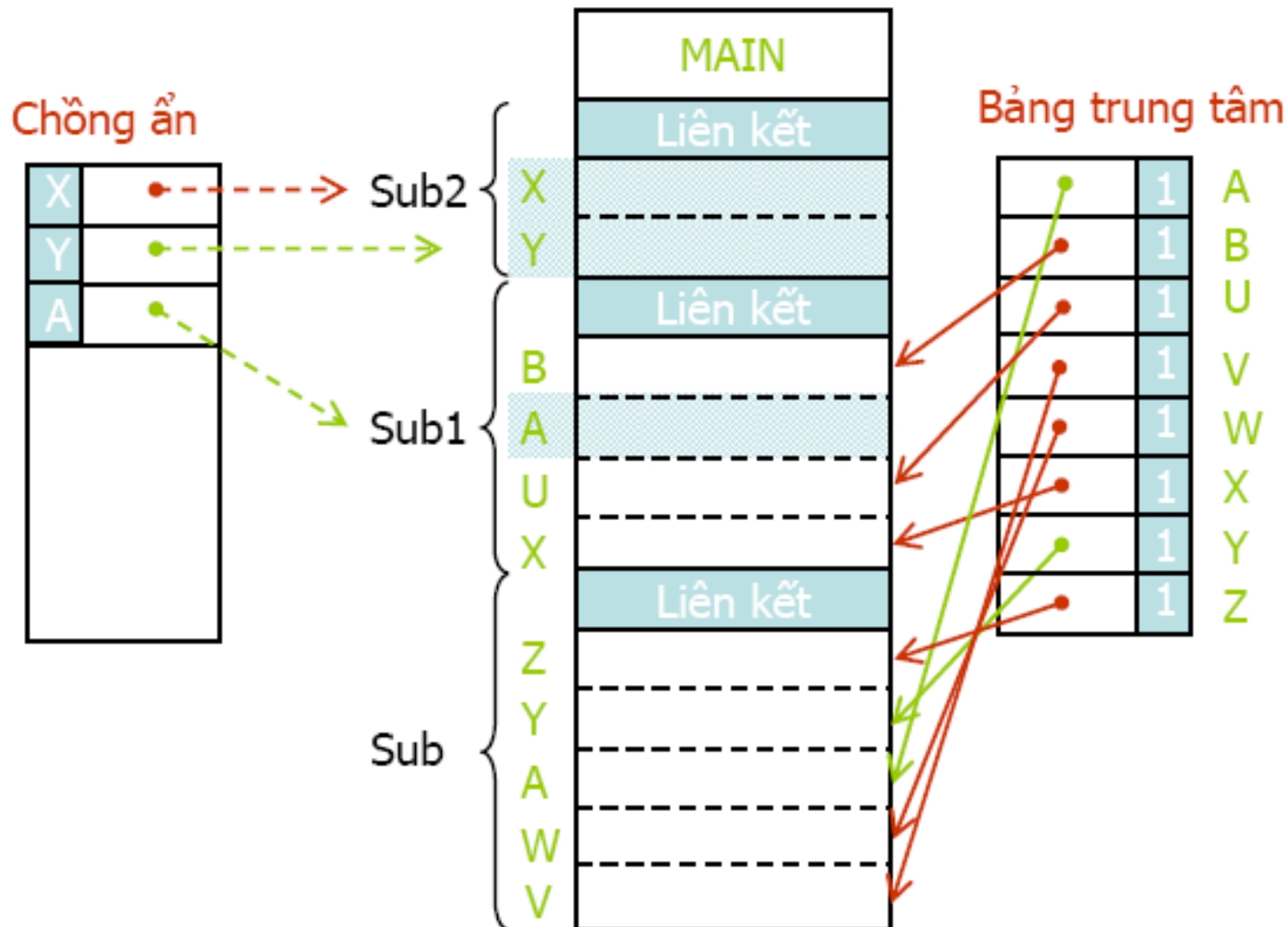
Bảng trung tâm

	0	A
	0	B
	0	U
	0	V
	0	W
	1	X
	1	Y
	0	Z

Hiện thực



Hiện thực



Dữ liệu dùng chung: Cấu trúc khối

- Quy tắc tầm vực tĩnh: slide 11
- Quy tắc tầm vực động: phù hợp với quy tắc tầm vực tĩnh

Ví dụ

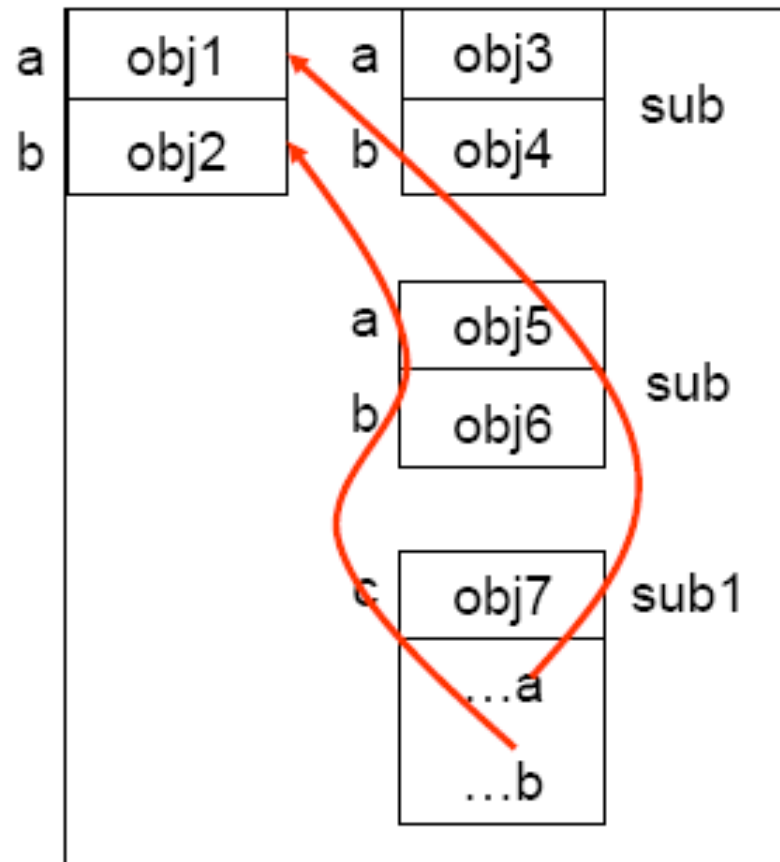
```

var a,b:integer
procedure sub ()
  var a,b:real;
  begin
    ...
  end
procedure sub1()
  var c:integer
  begin
    ...a...b...
  end
begin
  ...
end

```

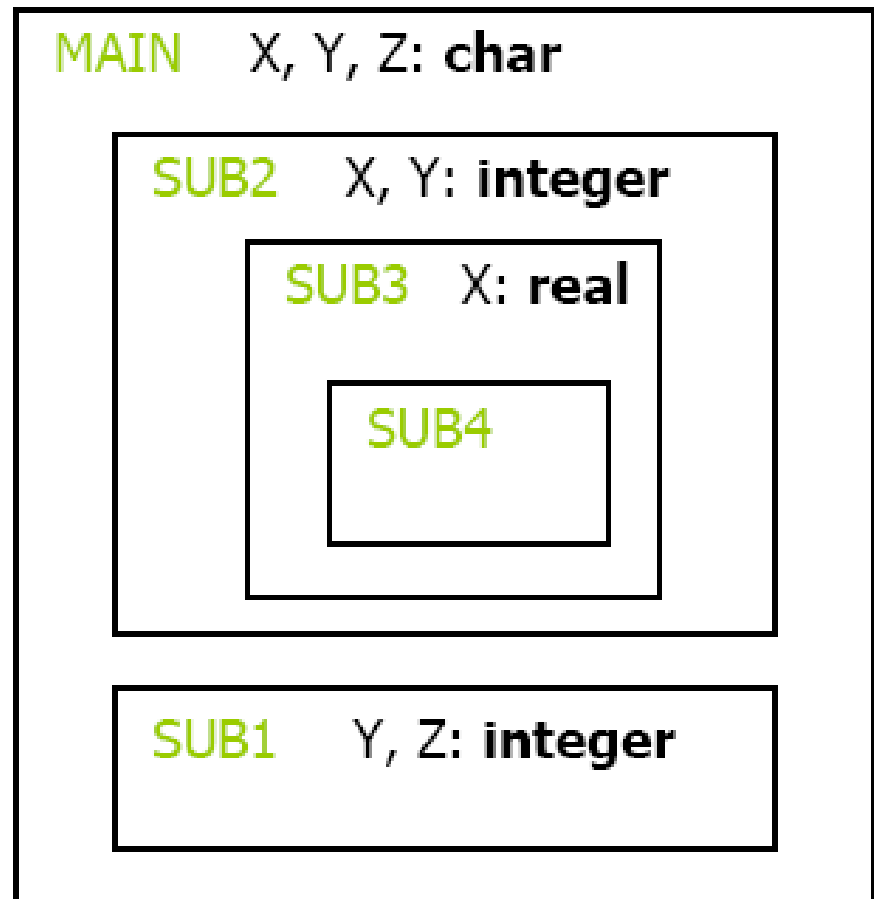
Tĩnh

Động: main → sub → sub → sub1



Hiện thực: Tĩnh

- Dùng bảng danh hiệu và chồng tầm vực
- Program MAIN;
 - var X, Y, Z: char;
 - ...
 - procedure SUB2;
 - var X, Y: integer;
 - ...
 - procedure SUB3;
 - var X: real;
 - ...
 - procedure SUB4;
 - ...
 - procedure SUB1;
 - var Y, Z: integer;
- ...

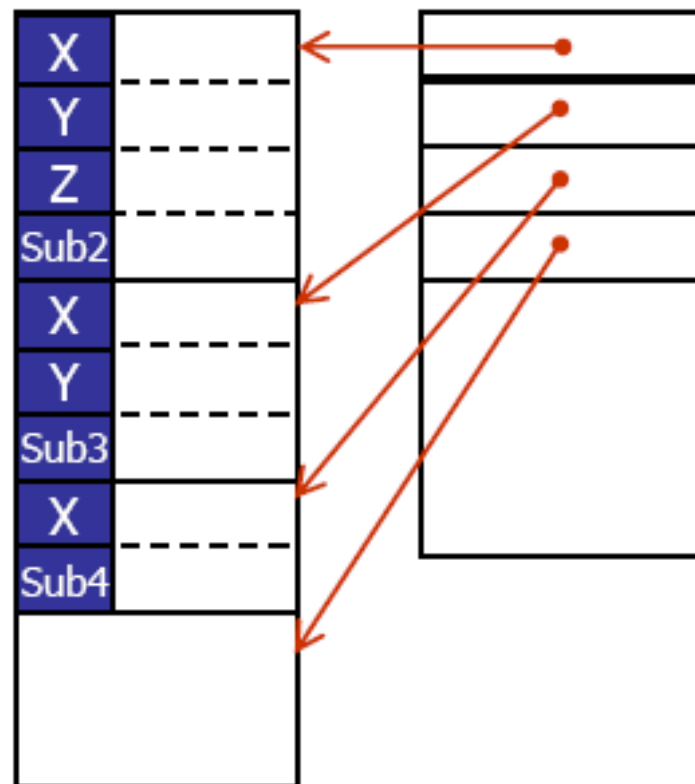


Hiện thực: Thời gian dịch

- program MAIN;
 - var X, Y, Z: char;
 - procedure SUB2;
 - var X, Y: integer;
 - procedure SUB3;
 - var X: real;
 - procedure SUB4;
 - begin...end
 - begin ...end
 - begin ...end
 - procedure SUB1;
 - var Y, Z: integer;
 - begin...end
- begin ...end

Bảng danh hiệu

Chồng tầm vực

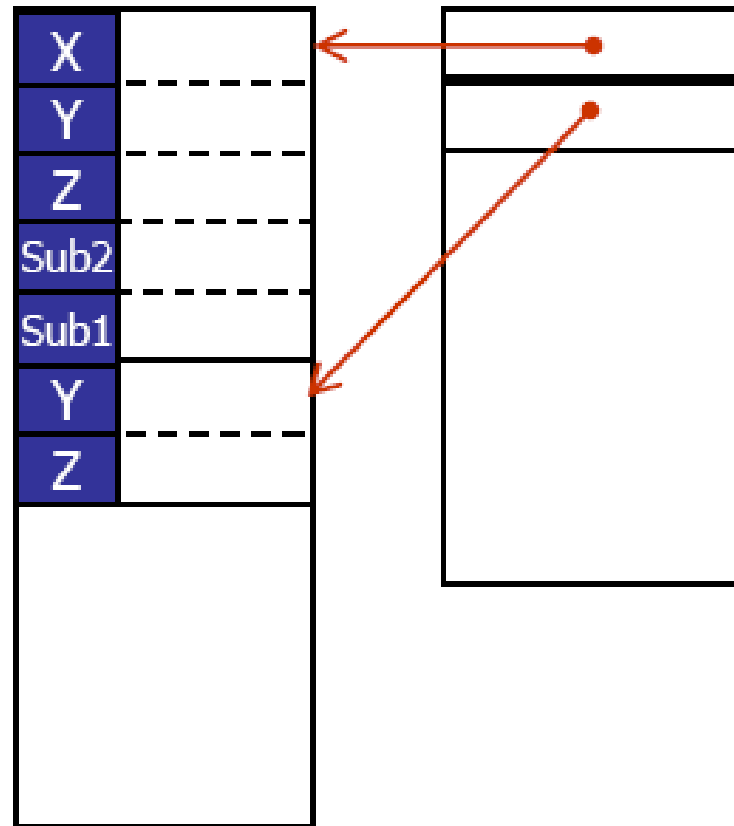


Hiện thực: Thời gian dịch

- Program MAIN;
 - var X, Y, Z: char;
 - procedure SUB2;
 - var X, Y: integer;
 - procedure SUB3;
 - var X: real;
 - procedure SUB4;
 - begin...end
 - begin...end
 - begin ...end
 - procedure SUB1;
 - var Y, Z: integer;
 - begin...end
- begin ...end

Bảng danh hiệu

Chồng tầm vực



Hiện thực: Thời gian chạy

- Program MAIN;
 - var X, Y, Z: char;
 - ...
 - procedure SUB2;
 - var X, Y: integer;
 - ...
 - procedure SUB3;
 - var X: real;
 - ...
 - procedure SUB4;
 - ...
 - procedure SUB1;
 - var Y, Z: integer;
 - ...

MAIN



SUB1



SUB2



SUB3

Ví dụ

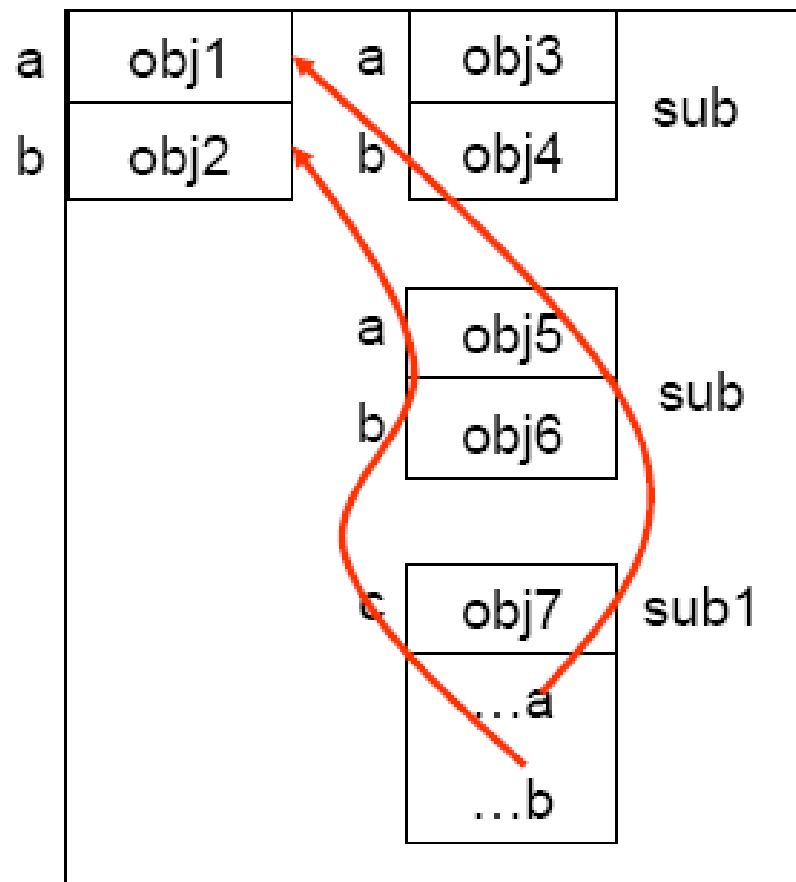
```

var a,b:integer
procedure sub ()
  var a:b:real;
  begin
    ...
  end
procedure sub1()
  var c:integer
  begin
    ...a...b...
  end
begin
  ...
end

```

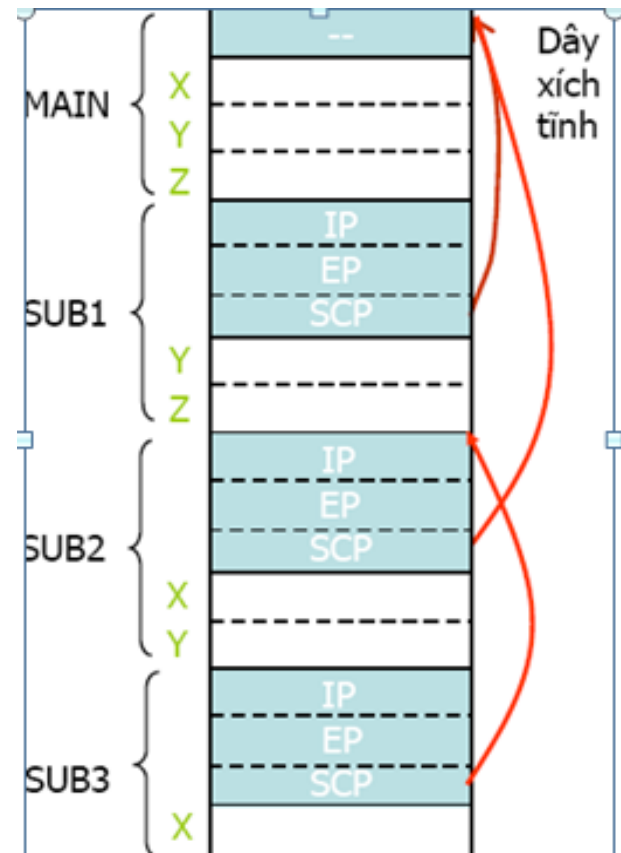
Tĩnh

Động: main → sub → sub → sub1



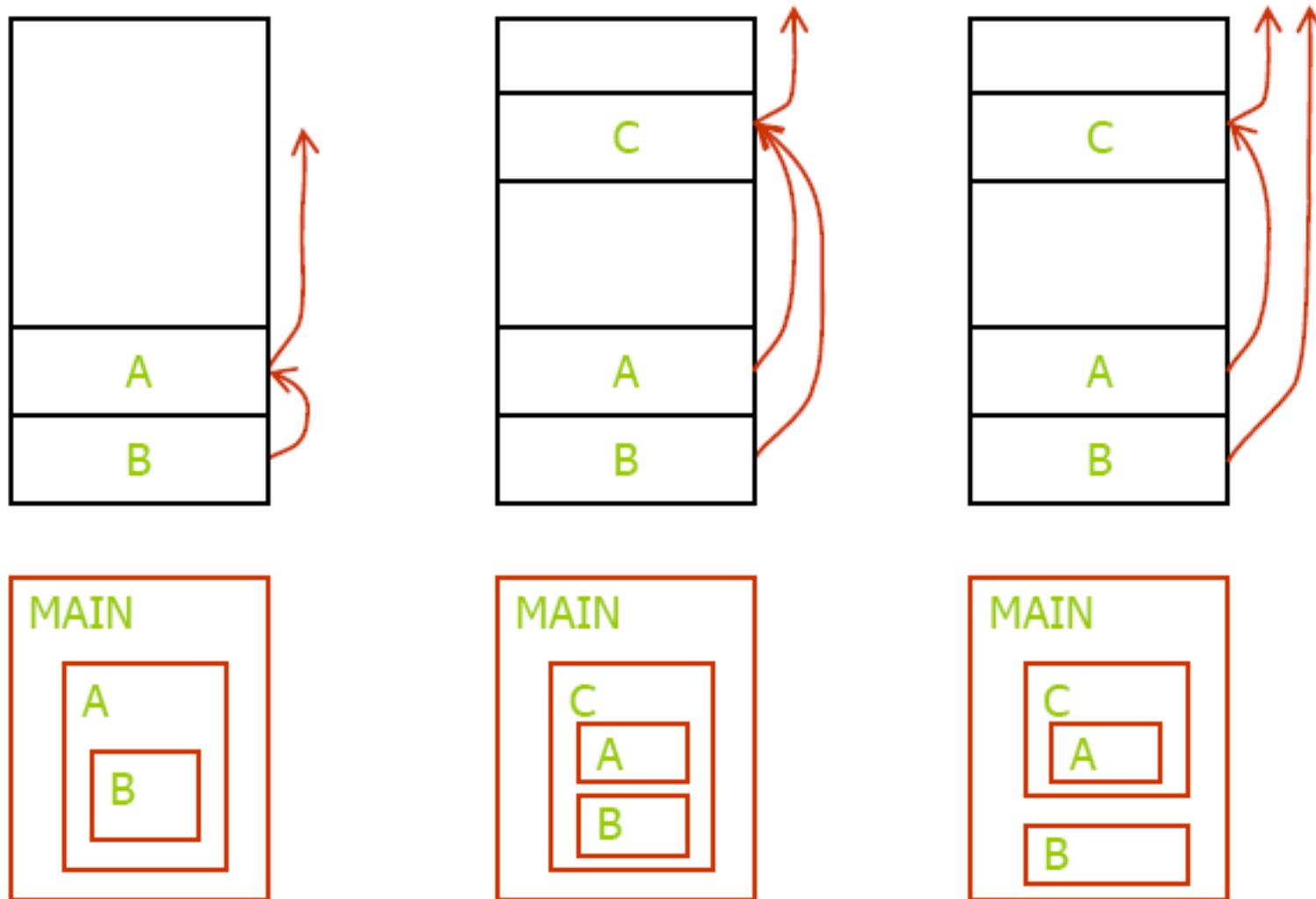
Hiện thực: Thời gian chạy

- Program MAIN;
 - var X, Y, Z: char;
 - procedure SUB2;
 - var X, Y: integer;
 - procedure SUB3;
 - var X: real;
 - procedure SUB4;
 - ...
 - procedure SUB1;
 - var Y, Z: integer;
 - ...



MAIN → SUB1 → SUB2 → SUB3

Hiện thực: Thời gian chạy



Hiện thực: Thời gian chạy

