

Ngôn ngữ lập trình C

B11: Con trỏ



PHENIKAA
UNIVERSITY

Khoa Công nghệ thông tin

Địa chỉ và giá trị của một biến

- Bộ nhớ như một dãy các byte nhớ.
- Các byte nhớ được xác định một cách duy nhất qua một địa chỉ.
- Biến được lưu trong bộ nhớ.
- Khi khai báo một biến
 - Chương trình dịch sẽ cấp phát cho biến đó một số ô nhớ liên tiếp đủ để chứa nội dung của biến. Ví dụ một biến số nguyên (int) được cấp phát 2 byte.
 - Địa chỉ của một biến chính là địa chỉ của byte đầu tiên trong số đó.

Địa chỉ và giá trị của một biến

- Một biến luôn có hai đặc tính:

- Địa chỉ của biến.
- Giá trị của biến.

- Ví dụ:

`- int i, j;`

`- i = 3;`

`- j = i + 1;`

Biến	Địa chỉ	Giá trị
i	FFEC	3
j	FFEE	4

Khái niệm và khai báo con trỏ

- Con trỏ là một biến mà giá trị của nó là địa chỉ của một vùng nhớ.
- Khai báo con trỏ: Cú pháp khai báo một con trỏ như sau:

```
Kieu_du_lieu *ten_bien_con_trỏ;
```

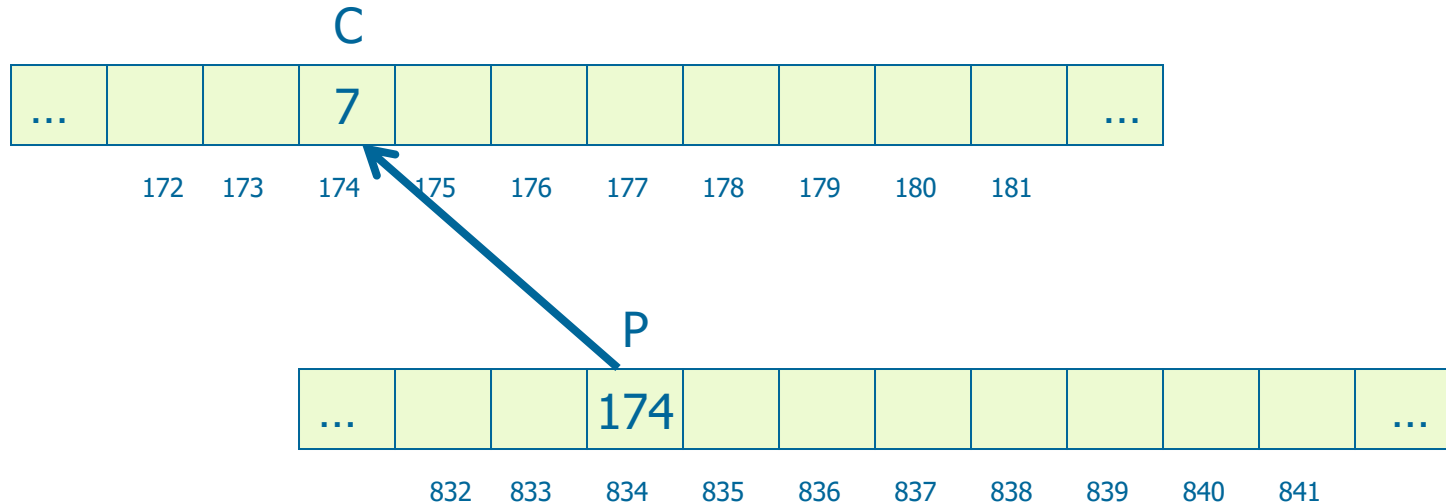
- Ví dụ:
 - `int i = 3;`
 - `int *p;`
 - `p = &i;`

Biến	Địa chỉ	Giá trị
i	FFEC	3
p	FFEE	FFEC

- Một con trỏ chỉ có thể trỏ tới một đối tượng cùng kiểu.

Con trỏ

- Như hình dưới: con trỏ **P** có giá trị là địa chỉ của biến **c** và trỏ tới biến **c**



Toán tử & và *

- **Toán tử &:** Trả về địa chỉ của biến.
- **Toán tử *:** Trả về giá trị chứa trong vùng nhớ được trỏ bởi con trỏ.
- Cả hai toán tử * và & có độ ưu tiên cao hơn tất cả các toán tử số học ngoại trừ toán tử đảo dấu.
- Ví dụ 11.1:

```
void main()  
{  
    int i = 3; int *p;  
    p = &i;  
    printf("*p = %d \n", *p);  
    printf("p = %p \n", p);  
}
```



```
*p = 3  
p = 0x7fff84f8572c
```

Sử dụng biến con trỏ

- Một biến con trỏ có thể được gán bởi: địa chỉ của một biến khác:

```
ten_bien_con_tro = &ten_bien;
```

- Giá trị của một con trỏ khác (tốt nhất là cùng kiểu):

```
ten_bien_con_tro2 = ten_bien_con_tro1;
```

- Giá trị NULL (số 0):

```
ten_bien_con_tro = 0; // hoặc là NULL
```

- Gán giá trị cho biến con trỏ:

```
*ten_bien_con_tro = 10;
```

Sử dụng biến con trỏ

Ví dụ: sử dụng biến con trỏ

```
void main()
{
    int i = 3, j = 6;
    int *p1, *p2;
    p1 = &i;
    p2 = &j;
    *p1 = *p2;
}
```

biến	địa chỉ	giá trị
i	FFEC	3
j	FFEE	6
p1	FFDA	FFEC
p2	FFDC	FFEE



biến	địa chỉ	giá trị
i	FFEC	6
j	FFEE	6
p1	FFDA	FFEC
p2	FFDC	FFEE

```
void main()
{
```

```
    int i = 3, j = 6;
    int *p1, *p2;
    p1 = &i;
    p2 = &j;
    p1 = p2;
```

```
}
```

biến	địa chỉ	giá trị
i	FFEC	3
j	FFEE	6
p1	FFDA	FFEC
p2	FFDC	FFEE



biến	địa chỉ	giá trị
i	FFEC	3
j	FFEE	6
p1	FFDA	FFEE
p2	FFDC	FFEE

Bài tập 11.1

- Viết chương trình nhập vào 3 số nguyên. Khai báo một con trỏ và trỏ đến lần lượt các số nguyên đó. Hiển thị giá trị của con trỏ trong mỗi lần chỏ đến một số.

Ví dụ 11.2

- Viết chương trình in ra địa chỉ (dạng số hexa) của một mảng được định nghĩa như dưới đây:

```
int a[7]= {13, -355, 235, 47, 67, 943, 1222} ;
```

```
#include<stdio.h>
void main(){
    int a[7]={13, -355, 235, 47, 67, 943, 1222};
    int i;
    for(i=0;i<7;i++)
        printf("a[%d]:\t%p\t%p\n",i,a+i,&a[i]);
}
```

Ví dụ 11.3

- Viết chương trình cung cấp giá trị của 3 số thực (float) vào các biến x , y , z . Sau đó cộng thêm 100 vào giá trị của mỗi biến bằng cách chỉ sử dụng con trỏ.

Đáp án ví dụ 11.3

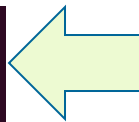
```
#include <stdio.h>
void main()
{
    float x = 25.1, y = 50.8, z = 75.5;
    float *ptr;
    printf("Here are the values of x, y, and z:\n");
    printf("%f %f %f\n", x, y, z);
    ptr = &x; // Store the address of x in ptr
    *ptr += 100; // Add 100 to the value in x
    ptr = &y; // Store the address of y in ptr
    *ptr += 100; // Add 100 to the value in y
    ptr = &z; // Store the address of z in ptr
    *ptr += 100; // Add 100 to the value in z
    printf("Once again, here are the values of x, y,
    and z:\n");
    printf("%f %f %f\n", x, y, z);
}
```

Truyền tham biến cho hàm

- Giải thích lại ví dụ 9.4:

```
#include<stdio.h>
void swap(int x, int y){
    int temp;
    printf("Hai so nguyen ban dau duoc truyen vao ham: %d, %d\n",x,y);
    printf("Ham thu hien viec trao doi hai so nguyen nay.\n");
    temp = x;
    x = y;
    y = temp;
    printf("Hai so nguyen sau khi duoc trao doi trong ham: %d, %d\n",x,y);
}

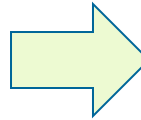
void main(){
    int x, y;
    printf("Moi ban nhap vao hai so nguyen:\n");
    printf("x = "); scanf("%d",&x);
    printf("y = "); scanf("%d",&y);
    printf("Hai so nguyen truoc khi duoc truyen vao ham swap: %d, %d\n",x,y);
    printf("Goi ham swap\n");
    swap(x,y);
    printf("Hai so nguyen sau khi thuc hien xong ham swap: %d, %d\n",x,y);
}
```



Không đổi giá trị của 2 biến sau khi gọi hàm xong

Biện pháp:

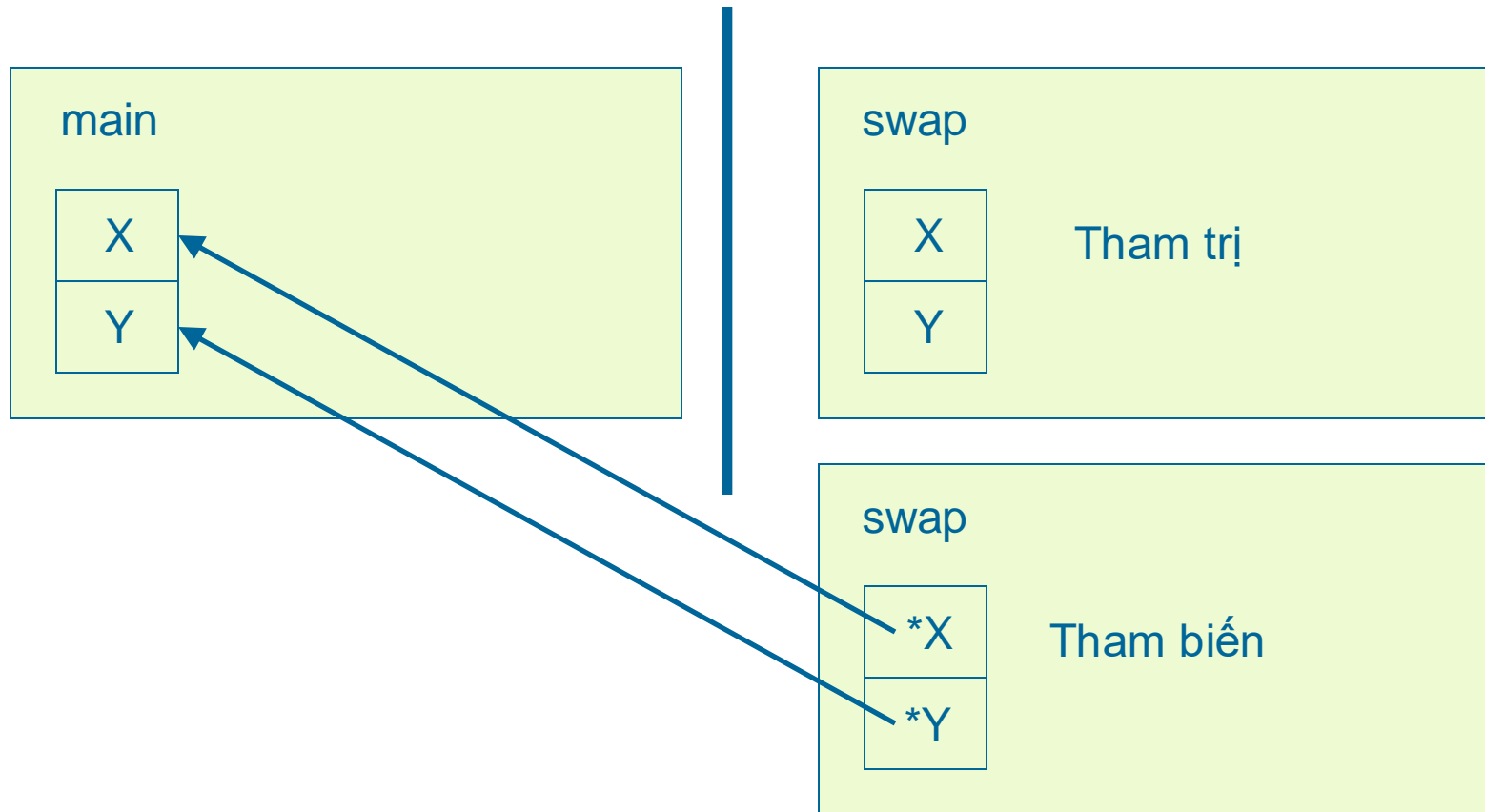
Đổi được giá trị của 2 biến sau khi gọi hàm xong.



```
#include<stdio.h>
void swap(int* x, int* y){
    int temp;
    printf("Hai so nguyen ban dau duoc truyen vao ham: %d, %d\n",*x,*y);
    printf("Ham thu hien viec trao doi hai so nguyen nay.\n");
    temp = *x;
    *x = *y;
    *y = temp;
    printf("Hai so nguyen sau khi duoc trao doi trong ham: %d, %d\n",*x,*y);
}

void main(){
    int x, y;
    printf("Moi ban nhap vao hai so nguyen:\n");
    printf("x = "); scanf("%d",&x);
    printf("y = "); scanf("%d",&y);
    printf("Hai so nguyen truoc khi duoc truyen vao ham swap: %d, %d\n",x,y);
    printf("Goi ham swap\n");
    swap(&x, &y);
    printf("Hai so nguyen sau khi thuc hien xong ham swap: %d, %d\n",x,y);
}
```

Truyền tham biến cho hàm



Bài tập 11.2

- Viết hàm lấy ba biến (a, b, c) thành các tham số riêng biệt và xoay vòng các giá trị được lưu trữ để giá trị của a thành b, b thành c và c thành a. Kiểm tra hoạt động của hàm này trong một chương trình.

Bài tập 11.3

- Khai báo 3 biến nguyên **x, y, z** và 3 biến con trỏ nguyên **p, q, r**. Thiết lập cho **x, y, z** nhận 3 giá trị riêng biệt. Thiết lập **p, q, r** trỏ tới địa chỉ của 3 biến **x, y, z** tương ứng.
 - In ra màn hình cùng với nhãn các giá trị của **x, y, z, p, q, r, *p, *q, *r**.
 - Tráo đổi giá trị của **x, y, z** với nhau. In ra màn hình cùng với nhãn các giá trị của **x, y, z, p, q, r, *p, *q, *r**.
 - Tráo đổi giá trị của **p, q, r**. In ra màn hình cùng với nhãn các giá trị của **x, y, z, p, q, r, *p, *q, *r**.

Bài tập

4. Chương trình C sử dụng con trỏ để duyệt một mảng các số nguyên và in các giá trị theo thứ tự đảo ngược.
5. Viết chương trình kiểm tra một chuỗi có phải là palindrome. Một chuỗi được gọi là palindrome nếu sau khi đảo ngược các ký tự của nó ta vẫn nhận được chuỗi như ban đầu. Ví dụ chuỗi MADAM.
6. Viết chương trình để nối 2 chuỗi bằng con trỏ.

Bài tập 11.7

- Viết chương trình quản lý lương nhân viên:
 - N nhân viên, lương hiện tại được lưu trong mảng có N phần tử. Mã nhân viên là số thứ tự mảng.
 - Mảng lưu năm bắt đầu làm việc.
 - Viết hàm để tăng lương cho từng nhân viên. Quy tắc tăng lương là cứ 3 năm tăng 1 lần, mỗi lần 1% mức hiện tại. Đầu vào của hàm là năm bắt đầu làm việc, mức lương hiện tại và đầu ra chính là mức lương đã được tăng (nếu có).
 - Yêu cầu: sử dụng con trỏ.