

CONFIDENTIAL

C Programming Introduction

week 11: Pointers

For HEDSPI Project

Địa chỉ bộ nhớ

- Mỗi phần tử nhớ trong bộ nhớ chiếm 1 byte. Mỗi byte đi với 1 số, là địa chỉ của nó.
- Hình dưới hiển thị các địa chỉ từ 924 đến 940



Địa chỉ bộ nhớ

- Phép toán 1 ngôi & cho địa chỉ biến

```
#include <stdio.h>
int main(){
    float fl=3.14;
    printf("fl's address=%u\n", (unsigned int) &fl);
    return 0;
}
```



Bài 12.1

- Viết chương trình nhập 3 số nguyên. Tạo 1 biến con trỏ để trữ lần lượt vào từng biến 1. Hiện giá trị trữ bởi con trỏ

Solution

```
#include <stdio.h>

int main(){
    int x, y, z;
    int* ptr;
    printf("Enter three integers: ");
    scanf("%d %d %d", &x, &y, &z);
    printf("\nThe three integers are:\n");
    ptr = &x;
    printf("x = %d\n", *ptr);
    ptr = &y;
    printf("y = %d\n", *ptr);
    ptr = &z;
    printf("z = %d\n", *ptr);
    return 0;
}
```

Bài 12.2

- Viết chương trình in ra địa chỉ (ở hệ 16) của 5 phần tử đầu tiên của mảng được định nghĩa dưới đây:

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

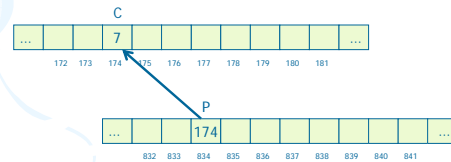
Khai báo biến con trỏ

```
type *variable_name;
```

- 1 con trỏ được khai báo bằng cách thêm dấu * trước tên biến.
- Con trỏ là 1 biến chứa địa chỉ ô nhớ
- Địa chỉ là địa chỉ của biến hoặc mảng ta định nghĩa

Con trỏ

- **ptr** là con trỏ trỏ đến địa chỉ của biến **c**

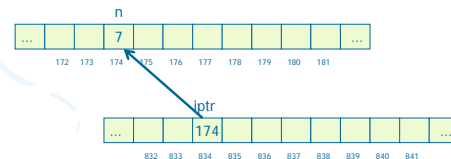


Tham chiếu

- Phép toán 1 ngôi **&** cho địa chỉ của biến
- Câu lệnh: **ptr = &c;**
- gán địa chỉ của biến con trỏ **ptr**, bây giờ **ptr** trỏ đến **c**
- Để in 1 con biến trỏ, sử dụng định dạng **%p**

Tham chiếu

```
int n;  
int *iptr; /* Declare P as a pointer to int */  
n = 7;  
iptr = &n;
```



Khử tham chiếu

- Phép toán 1 ngôi ***** là phép khử tham chiếu
- Dùng cho con trỏ
- Truy cập vào đối tượng được trỏ đến
- Câu lệnh: ***iptr = 5;**
đưa vào biến **n** (biến trỏ bởi **iptr**) giá trị 5

Bài 12.3

- Viết chương trình nhập 3 giá trị thực a, b, c. Sau đó + 100 vào từng phần tử đó chỉ dùng 1 biến trỏ.

Truyền tham số dưới dạng giá trị

- Các hàm đã dùng chỉ truyền giá trị
- Các hàm này có thể thay đổi giá trị được truyền vào
- Nhưng khi ra khỏi hàm, các tham số truyền vào này không thay đổi giá trị

Hàm đổi chỗ SAI

- Hàm đổi chỗ, nếu truyền vào giá trị, sẽ không thể đổi giá trị các biến truyền vào

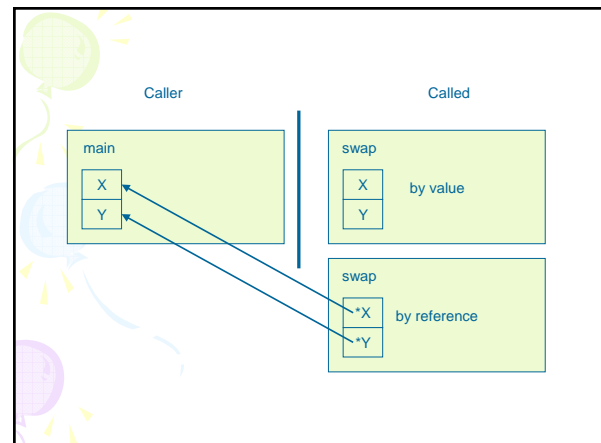
```
void swap(int x, int y)
{
    int tmp = x;
    x = y;
    y = tmp;
}
```

Sửa?

- Truyền các tham số cho hàm dưới dạng con trỏ thay vì giá trị

```
void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}
```

- Lời gọi trong chương trình: `swap(&x, &y);`



Bài 12.4

- Viết hàm nhận vào 3 biến (a, b, c) in và quay tròn 3 giá trị này (a=b, b=c, c=a).
- Thử nghiệm hàm này trong chương trình

Bài 12.5

Khai báo các biến `int x, y, z` và `int* p, q, r`. Gán `x, y, z` 3 giá trị khác nhau. Đặt `p, q, r` là địa chỉ của 3 biến này.

- 1) In giá trị của `x, y, z, p, q, r, *p, *q, *r`.
- 2) Đổi chỗ các giá trị của `x, y, z`. In giá trị của `x, y, z, p, q, r, *p, *q, *r`.
- 3) Đổi chỗ các giá trị của `p, q, r`. In giá trị của `x, y, z, p, q, r, *p, *q, *r`.



Bài 12.6

- Để tăng lương cho 1 công nhân, viết hàm *incomeplus* dựa trên lương hiện tại và tổng số năm công tác (cần phải >3) tính từ thời điểm nhận lương hiện tại
- Thử nghiệm nó trong chương trình