

# BUỔI 17: CON TRỎ – FULL HOUSE

## 1. Địa chỉ biến trong C.

### a. Toán tử địa chỉ: &

Khi chúng ta sử dụng một dữ liệu như biến, hàm hoặc struct trong chương trình C, một vùng trong bộ nhớ máy tính sẽ được chuẩn bị để lưu tạm thời **giá trị** của dữ liệu. Để truy cập vào dữ liệu này, bạn cần phải biết **vị trí** của vùng lưu giữ nó trong bộ nhớ.

Giống như số nhà của bạn được đăng ký trên phường, trong máy tính mỗi dữ liệu sẽ được máy tính cấp phát cho một số để biểu thị **vị trí** của nó trong bộ nhớ của máy tính. Số này chỉ **điểm đầu của vùng chứa dữ liệu** trong bộ nhớ, và nó sẽ đại diện cho cả vùng đó. Và chúng ta gọi số chỉ vị trí này là **địa chỉ của dữ liệu** trong chương trình.

```
1 #include <stdio.h>
2
3 int main(){
4     int n;
5     scanf("%d",&n);
6     printf("Gia tri cua bien n: %d",n);
7     printf("\nDia chi cua bien n: %d",&n);
8 }
9
```

### b. Toán tử giá trị: \*

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d",&n);
    printf("Gia tri cua bien n: %d",n);
    //toan tu dia chi &
    printf("\nDia chi cua bien n: %d",&n);
    //Toan tu gia tri *
    printf("\nGia tri tai gia tri &n: %d",&n);
}
```

### c. Thay đổi giá trị thông qua địa chỉ

```
24 #include <stdio.h>
25
26 int main(){
27     int n = 15;
28     printf("Giá trị của biến n: %d",n);
29     //toán tử địa chỉ &
30     printf("\nĐịa chỉ của biến n: %d",&n);
31     //Toán tử giá trị *
32     printf("\nGiá trị tại địa chỉ &n: %d",&n);
33     //==> *&n == n
34     *&n += 100;
35     printf("\nGiá trị của biến n: %d",n);
36 }
```

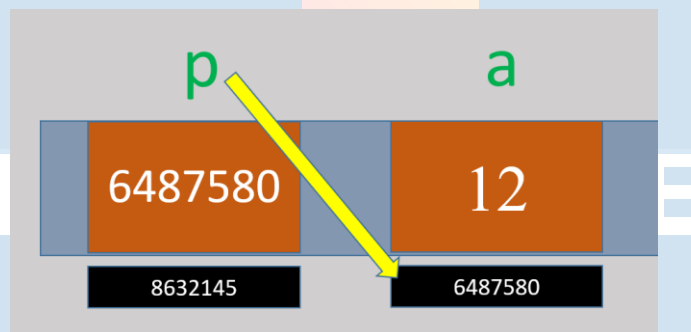
## 2. Con trỏ

- a. **Định nghĩa:** Con trỏ trong C cũng chỉ là là **biến**, cũng có thể khai báo, khởi tạo và lưu trữ giá trị và có địa chỉ của riêng nó. Nhưng biến con trỏ không lưu giá trị bình thường, nó là biến **trỏ tới 1 địa chỉ khác**, tức **mang giá trị là 1 địa chỉ**.

Chúng ta cùng thống nhất 1 số khái niệm khi làm việc với con trỏ nhé:

- Giá trị của con trỏ: địa chỉ mà con trỏ trỏ đến.
- Địa chỉ của con trỏ: địa chỉ của bản thân biến con trỏ đó.
- Giá trị của biến nơi con trỏ đang trỏ tới.
- Địa chỉ của biến nơi con trỏ đang trỏ tới = giá trị của con trỏ.

Chính vì con trỏ mang địa chỉ, nó là 1 biến đặc biệt có thêm những quyền năng mà biến bình thường không có. Nhờ việc nó mang địa chỉ, nó có thể trỏ lung tung trong bộ nhớ. Đây là 1 điểm mạnh nếu ta khai thác tốt nhưng nếu quản lý không tốt thì lại là 1 tai hại.



- b. Khai báo:

`type *name;`

Ví dụ 1:

```
int *p;
float *he_k;
void *jj4;
```

### Ví dụ 2:

```
int main(){
    int *p1;
    int * p2;
    int* p3;
}
```

```
#include <stdio.h>
```

```
int main(){
    int* p1; //khai bao bien p1 co kieu con tro int (int*)
    int *p3; //khai bao con tro p3 co kieu du lieu int
    int *p4, *p5, a,b;
    int* p6,c,d; //chi co p6 moi co kieu con tro
```

### c. Gán giá trị:

#### Ví dụ 1:

```
int main(){
    int *p;
    int n=10;
    p = &n;
    printf("%d",*p);
}
```

#### Ví dụ 2:

```
int n = 10;
int *p = &n;
```

#### Lưu ý:

- Con trỏ khi khai báo nên được khởi tạo giá trị ngay.
- Con trỏ kiểu **void** là loại biến con trỏ tổng quát, nó có thể nhận địa chỉ của biến bất kỳ ở bất cứ kiểu dữ liệu nào.

```
int main(){
    int n = 5;
    float *p = &n;
```

mpiler (2)			Resources	Compile Log	Debug	Find Results	Close
Col	File	Message					
	D:\Full House\C\GiaoAn\Li_Thuyet\Code buoi 17.cpp	In function 'int main()':					
16	D:\Full House\C\GiaoAn\Li_Thuyet\Code buoi 17.cpp	[Error] cannot convert 'int*' to 'float*' in initialization					

```
int main(){
    int n = 5;
    void *p = &n;
}
```

Process exited after 0.2261 seconds  
Press any key to continue . . .

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\Full H
- Output Size: 149,461914062
- Compilation Time: 0,17s

- Khởi tạo con trỏ bằng địa chỉ **NULL** nếu chưa cần dùng theo cách sau: **int \*p = NULL**. Khi đó con trỏ **NULL** luôn có giá trị 0.

#### d. Một số thao tác với con trỏ

- Tổng quát

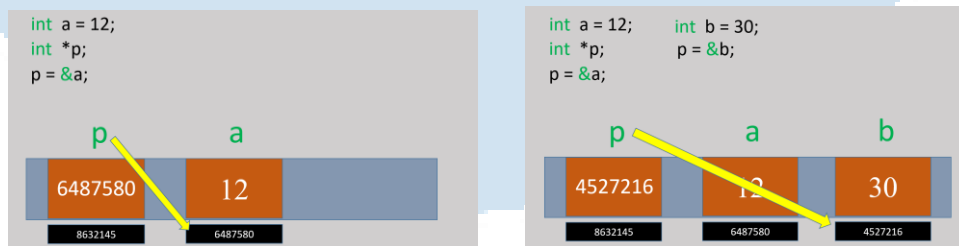
```
#include <stdio.h>

int main(){
    int n = 5;
    int *p = &n;

    printf("Gia tri cua n: %d",n);
    printf("\nDia chi cua n: %d",&n);
    printf("\nGia tri cua p: %d",p);
    printf("\nGia tri con tro p tro toi: %d",*p);

    *p += 100;
    printf("\nGia tri moi cua n: %d",n);
}
```

- Thay đổi giá trị của con trỏ.



```

64 #include <stdio.h>
65
66 int main(){
67     int n = 5;
68     int *p = &n;
69
70     printf("Gia tri cua n: %d",n);
71     printf("\nDia chi cua n: %d",&n);
72     printf("\nGia tri cua p: %d",p);
73     printf("\nGia tri con tro p tro toi: %d",*p);
74
75     *p += 100;
76     printf("\nGia tri moi cua n: %d",n);
77
78     int m = 24;
79     p = &m;
80     printf("\nGia tri cua a: %d",n);
81 }

```

- 23.02 - Hoán đổi

```
#include <stdio.h>
```

```

int main(){
    int a,b;
    scanf("%d%d",&a,&b);
    int *p1 = &a;
    int *p2 = &b;

    int c= *p1;
    *p1 = *p2;
    *p2 = c;

    printf("%d %d",a,b);
}

```

### 3. Con trỏ hàm

- a. **Định nghĩa:** Con trỏ **lưu trữ địa chỉ của một hàm** là con trỏ hàm trong C, và sử dụng nó để truy cập vào địa chỉ của hàm, cũng như thực thi các xử lý bên trong hàm đó.

b. **Khai báo con trỏ hàm trong C:**

`type (*fp) (para_type_1, para_type_2, para_type_3...);`

- type: kiểu dữ liệu
- fb: Tên con trỏ
- para\_type: là kiểu của các đối số (parameter) được truyền vào hàm.

```
void (*po1)(int, int);
```

```
int (*po2)(int);
```

```
void (*po3)();
```

c. **Gán địa chỉ của hàm cho con trỏ hàm và gọi hàm**

Cú pháp gán: `fp = funct;`

Cú pháp gọi: `(*fp) (para_1, para_2, para_3,...);`

```
#include <stdio.h>

int sum(int a, int b){
    return a + b;
}

int main(void){
    int n=3, m=5;
    int (*fp)(int, int);

    fp = sum;        // gán địa chỉ con trỏ vào hàm
    int s = (*fp) (n,m); //gọi hàm
    printf("%d",s);
}
```