



Nguyễn Phúc Khải

# CHƯƠNG 13: *Kiểu dữ liệu có cấu trúc và kiểu dữ liệu tự định nghĩa*





# Các nội dung:

- Kiểu STRUCT
- Kiểu UNION
- Kiểu ENUM (*Enumerated*)
- Định nghĩa kiểu bằng TYPEDEF





- **Khái niệm - Khai báo struct**
- Struct (tạm dịch là cấu trúc) là một kiểu dữ liệu phức hợp được tạo từ các kiểu dữ liệu khác, các kiểu dữ liệu này được sử dụng khai báo cho các biến thành phần của biến kiểu struct.

**struct tên\_cấu\_trúc**

{

Khai báo các biến thành phần

};



■ Ví dụ:

**struct** sinh\_vien

{

char ma\_so[10];

char ho\_ten[40];

int tuoi;

char dia\_chi[80];

};



- Cú pháp của một khai báo biến cấu trúc giống như khai báo biến bình thường:

**struct tên\_struct tên\_biến;**

- Ví dụ:

```
struct sinh_vien sv1, sv2;
```

```
struct sinh_vien sv1 = { "49508XX", "Tran van  
V", 21, "42 Tr Dinh p.13 q.TB"};
```



## ■ Ví dụ:

struct sinh\_vien

{

char ma\_so[10];

char ho\_ten[40];

int tuoi;

char dia\_chi[80];

}

sv1, sv2;



10 byte	40 byte	2 byte	80 byte
ma_so	ho_ten	tuoi	dia_chi



- Để truy xuất một thành phần của biến cấu trúc, C có toán tử chấm “.” để lấy từng thành phần.
- Ví dụ:

```
strcpy (sv1.ma_so, "49508XX");
```

```
strcpy (sv1.ho_ten, "Tran van D");
```

```
sv1.tuoi = 21;
```

```
strcpy (sv1.dia_chi, " 42 Tr Dinh p.13 q.TB");
```





- C cho phép gán các cấu trúc cùng kiểu cho nhau qua tên biến cấu trúc thay vì phải gán từng thành phần cho nhau.
- Ví dụ:

`sv2 = sv1;`





- Các thành phần của biến struct cũng là biến bình thường, nên ta có thể lấy địa chỉ của chúng.
- Kiểu struct có thể được lấy kích thước tính theo byte nhờ toán tử sizeof.
- Ví dụ: **sizeof** (struct sinh\_vien);



- **Mảng các struct:**
- Cú pháp khai báo mảng các struct:  
**struct** ten\_cau\_truc ten\_mang [kich\_thuoc];
- Ví dụ:

**struct** sinh\_vien sv[50];



- **Pointer trở tới một struct:**
- Cú pháp khai báo biến pointer này như sau:

**struct** tên\_cấu\_trúc **\*tên\_pointer**;

- Ví dụ:

```
struct sinh_vien a, sv[50], *pa, *psv;
```

```
pa = &a;
```

```
psv = sv;
```



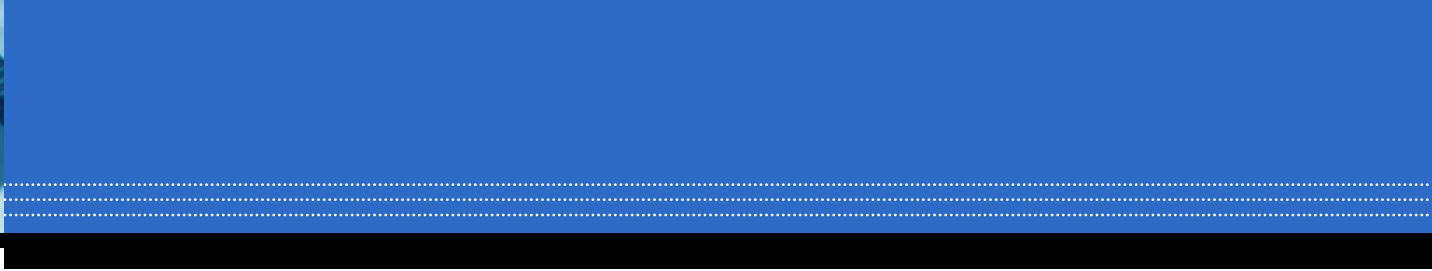
- Việc truy xuất đến một thành phần của một cấu trúc thông qua một pointer được thực hiện bằng toán tử lấy thành phần của đối tượng của pointer, ký hiệu là  $\rightarrow$  (có thể gọi là toán tử mũi tên).

- Ví dụ:

```
printf ("Ho ten sinh vien: %s \n", psv  $\rightarrow$  ho_ten);
```

- Hay

```
printf ("Ho ten sinh vien: %s \n", (*psv).ho_ten);
```





- **Struct dạng field:**
- C cho phép ta khai báo các thành phần của **struct** theo bit hoặc một nhóm bit. Một thành phần như vậy được gọi là một field (tạm dịch là vùng).

**struct** tên\_cấu\_trúc

```
{  kiểu tên_vùng 1: số_bit1;  
    kiểu tên_vùng 2: số_bit2;  
  
    ...  
}
```

- Với **kiểu** chỉ có thể là unsigned, signed hoặc int



## ■ Ví dụ: struct date

{

unsigned day: 5;  
unsigned month: 4;  
unsigned year: 6;  
int: 0;

} ngày;





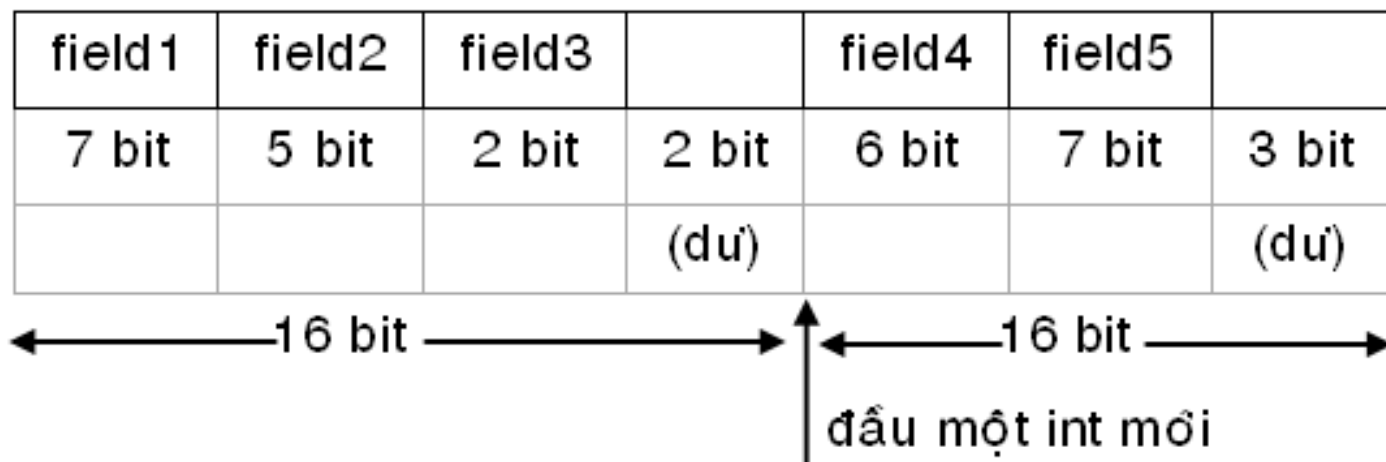


## ■ Struct dạng field:

- Mỗi vùng chỉ có thể dài tối đa 16 bit (một int) và được cấp chỗ trong một int, chứ không thể nằm trên hai int khác nhau được.
- Sự phân bố bit cho các field trong một int của struct (từ trái sang phải hay ngược lại), không phân biệt được.
- Mọi thao tác thực hiện trên biến kiểu field có liên quan đến địa chỉ đều không được thực hiện



## ■ Ví dụ:



struct vi\_du

```
{    unsigned field1: 7;
    unsigned field2: 5;
    unsigned field3: 2;
    unsigned field4: 6;
    unsigned field5: 7;
    } vd;
```



# *KIỂU UNION*

- Ngôn ngữ C có kiểu dữ liệu *union* (tạm dịch là kiểu hợp nhất), đây là một kiểu dữ liệu đặc biệt mà nếu được khai báo thì ứng với một vùng nhớ, giá trị ở mỗi thời điểm khác nhau thì có thể có kiểu khác nhau tùy vào việc sử dụng biến thành phần trong nó.





- Khai báo biến kiểu union:

**union** tên\_union

{

    khai\_báo\_biến\_thành\_phần

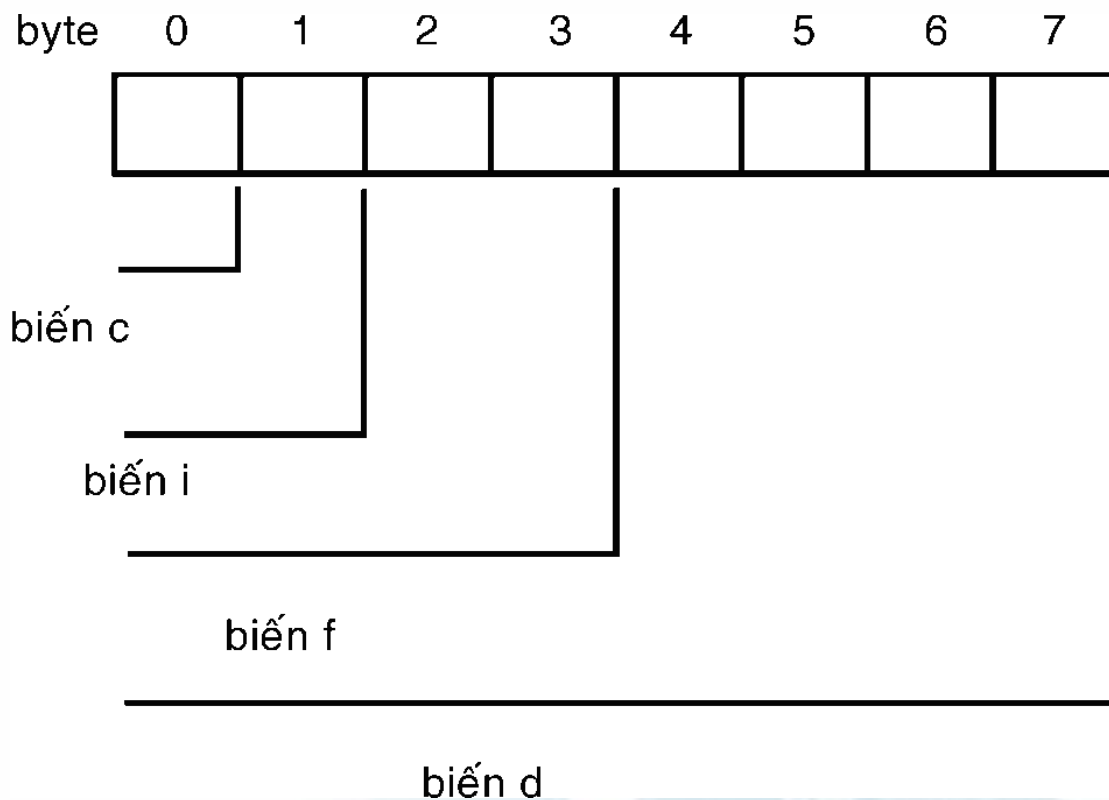
} biến, biến [...];

hoặc

**union** tên\_union biến, biến [...];



■ Ví dụ:  
union thu  
{  
    char c;  
    int i;  
    float f;  
    double d;  
};





- Để truy xuất đến một biến thành phần của biến thuộc kiểu union, ta cũng dùng toán tử chấm “.”.
- Ví dụ:  
union thu a;  
a.c = 'a';
- Ta có thể khai báo một biến pointer chỉ đến một biến kiểu union.
- Ví dụ:  
union thu \*pthu, a;  
pthu = &a;



- Việc truy xuất đến một thành phần của union qua pointer được thực hiện bằng toán tử mũi tên, để lấy thành phần của union đang được pointer chỉ đến.
- Ví dụ: `pthu->c = 'A';`
- Kiểu union có thể được lấy kích thước tính theo byte qua toán tử `sizeof`



# Kết thúc chương 13