

# LẬP TRÌNH CĂN BẢN

---

## KIỂU CẤU TRÚC

Bộ môn Công nghệ phần mềm  
Khoa CNTT&TT, ĐH Cần Thơ





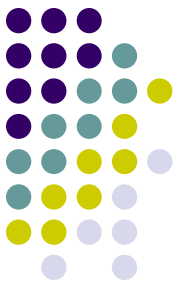
# Nội dung chương này

- Kiểu cấu trúc trong C
- Các thao tác trên biến kiểu cấu trúc
- Con trỏ và cấu trúc

# Kiểu cấu trúc trong C



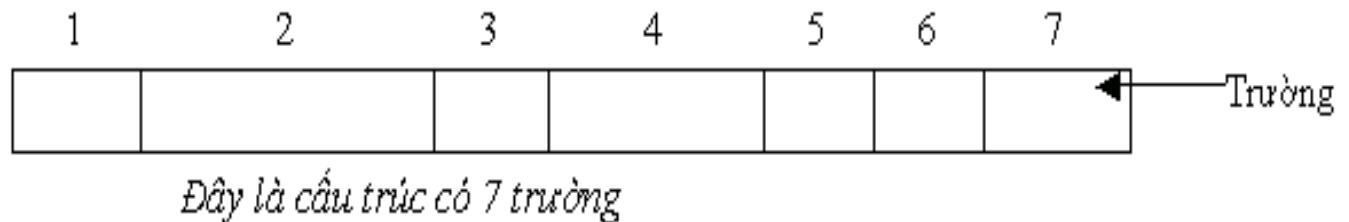
- Khái niệm
- Định nghĩa kiểu cấu trúc
- Khai báo biến cấu trúc



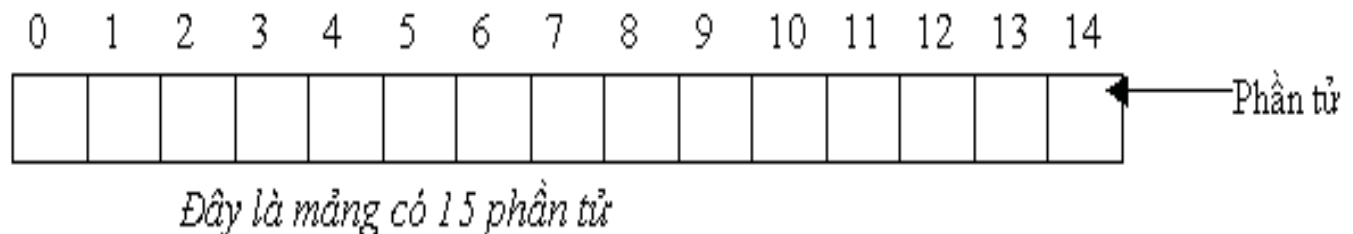
# Khái niệm

- Kiểu cấu trúc (struct) là kiểu dữ liệu bao gồm nhiều thành phần có kiểu khác nhau, mỗi thành phần được gọi là một trường (field)
- Nó khác với kiểu mảng (nơi mà các phần tử có cùng kiểu)
- Ví dụ:

**1 struct:**



**1 mảng:**



# Định nghĩa kiểu cấu trúc + Khai báo biến cấu trúc (1)

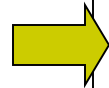


- Cách 1:

```
struct <Tên cấu trúc> {  
    <Kiểu> <Trường 1>;  
    <Kiểu> <Trường 2>;  
    .....  
    <Kiểu> <Trường n>;  
} [biến 1, biến 2];
```

## Ví dụ:

```
struct NgayThang{  
    unsigned char Ngay;  
    unsigned char Thang;  
    unsigned int Nam;  
};
```



```
struct SinhVien{  
    char MSSV[10];  
    char HoTen[40];  
    struct NgayThang NgaySinh;  
    int Phai;  
    char DiaChi[40];  
};
```

## ⇒ Khai báo biến:

```
struct NgayThang NgaySinh;  
struct SinhVien SV;
```

```
struct <Tên cấu trúc>   tên_biến;
```

# Định nghĩa kiểu cấu trúc + Khai báo biến cấu trúc (2)



- **Chú ý:**

- struct không tên:

```
struct {  
    int x;  
    int y;  
} A, B;
```

A và B là các struct có 2 thành phần x và y.

struct này không có tên, nên ngoài A và B, ta không thể định nghĩa thêm các biến khác được.

- Tuy nhiên

```
struct point {  
    int x;  
    int y;  
} A, B;
```

A và B là các biến có kiểu struct point.

Sau này ta có thể khai báo thêm các biến khác có kiểu struct point này.

# Định nghĩa kiểu cấu trúc + Khai báo biến cấu trúc (3)



- **Cách 2:**

```
typedef struct {  
    <Kiểu> <Trường 1> ;  
    <Kiểu> <Trường 2> ;  
    .....  
    <Kiểu> <Trường n> ;  
} <Tên cấu trúc>;
```

## Ví dụ:

```
typedef struct {  
    unsigned char Ngay;  
    unsigned char Thang;  
    unsigned int Nam;  
} NgayThang;
```

```
typedef struct {  
    char MSSV[10];  
    char HoTen[40];  
    NgayThang NgaySinh;  
    int Phai;  
    char DiaChi[40];  
} SinhVien;
```

## ⇒ Khai báo biến:

```
NgayThang NgaySinh;  
SinhVien SV;
```

```
<Tên cấu trúc>  tên_biến;
```

# Các thao tác trên biến kiểu cấu trúc



- Truy xuất đến từng trường của biến cấu trúc
- Khởi tạo cấu trúc



# Truy xuất đến từng trường (field) của biến cấu trúc (1)



- **Cú pháp:** <Biến cấu trúc>.<Tên trường>
- **Ví dụ 1:** Chương trình cho phép đọc dữ liệu từ bàn phím cho biến mẫu tin SinhVien và in biến mẫu tin đó lên màn hình:

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

typedef struct{
    unsigned char Ngay;
    unsigned char Thang;
    unsigned int Nam;
} NgayThang;
```



```
typedef struct{
    char MSSV[10];
    char HoTen[40];
    NgayThang NgaySinh;
    int Phai;
    char DiaChi[40];
} SinhVien;
```



# Truy xuất đến từng trường (field) của biến cấu trúc (2)



```
// Hàm in lên màn hình 1 mẫu tin SinhVien
void InSV(SinhVien s){
    printf("MSSV:   | Ho va ten   | Ngay Sinh   | Dia chi\n");
    printf("%s   | %s | %d-%d-%d | %s\n", s.MSSV, s.HoTen,
        s.NgaySinh.Ngay, s.NgaySinh.Thang, s.NgaySinh.Nam, s.DiaChi);
}

int main(){
    SinhVien SV;
    printf("Nhap MSSV: "); gets(SV.MSSV);
    printf("Nhap Ho va ten: "); gets(SV.HoTen);
    printf("Sinh ngay: "); scanf("%d", &SV.NgaySinh.Ngay);
    printf("Thang: "); scanf("%d", &SV.NgaySinh.Thang);
    printf("Nam: "); scanf("%d", &SV.NgaySinh.Nam);
    printf("Gioi tinh (0: Nu), (1: Nam): "); scanf("%d", &SV.Phai);
    fflush();
    printf("Dia chi: "); gets(SV.DiaChi);
    InSV(SV);
    SinhVien s;
    s=SV;        // Gán trị cho mẫu tin s
    InSV(s);
    getch();
    return 0;
}
```

# Truy xuất đến từng trường (field) của biến cấu trúc (3)



- Kết quả của 1 lần nhập:

```
TC.EXE
Nhap MSSV: 1040393
Nhap Ho va ten: Lam Nhat Tien
Sinh ngay: 29
Thang: 8
Nam: 1986
Gioi tinh (0: Nu), (1: Nam): 1
Dia chi: 1 Ly Tu Trong
MSSV:      |      Ho va ten      |      Ngay Sinh      |      Dia chi
1040393    |      Lam Nhat Tien    |      29-8-1986      |      1 Ly Tu Trong
MSSV:      |      Ho va ten      |      Ngay Sinh      |      Dia chi
1040393    |      Lam Nhat Tien    |      29-8-1986      |      1 Ly Tu Trong
```

# Truy xuất đến từng trường (field) của biến cấu trúc (4)



- **Lưu ý:**

- Các biến cấu trúc có thể gán cho nhau

- Ví dụ:

`s=SV; // gán để lấy giá trị toàn bộ cấu trúc`

- Ta không thể thực hiện được các thao tác sau đây cho biến cấu trúc:
  - Sử dụng các hàm xuất nhập trên biến cấu trúc
  - Các phép toán quan hệ, các phép toán số học và logic



# Khởi tạo cấu trúc

- Biến cấu trúc có thể được khởi tạo giá trị ban đầu lúc khai báo
- Ví dụ:

```
struct NgayThang NgaySinh = {29, 8, 1986};
```

```
struct point A = { 3, 4 };      tương đương      { struct point A;  
                                                                                 A.x = 3;  
                                                                                 A.y = 4;
```



# Con trỏ và cấu trúc

- Khai báo
- Sử dụng các con trỏ kiểu cấu trúc
- Truy cập các thành phần của cấu trúc đang được quản lý bởi con trỏ



# Khai báo (1)

- **Cú pháp:**

`struct` <Tên cấu trúc> \* <Tên biến con trỏ>;

- **Ví dụ 1:**

`struct` NgayThang \*p;

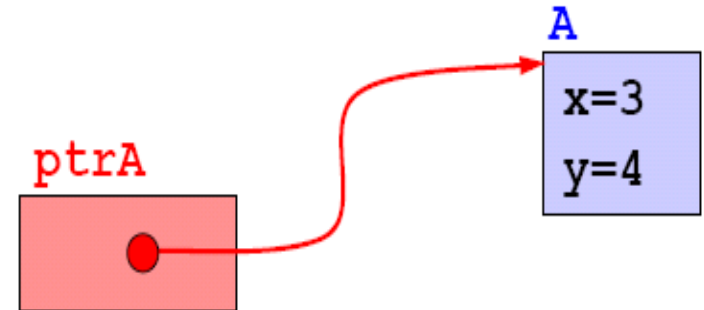
*hoặc*

NgayThang \*p;      // Nếu có dùng typedef

# Khai báo (2)

- Ví dụ 2:

```
struct point A = { 3, 4 };  
struct point* ptrA = &A;
```



- Truy cập đến các trường:

```
A.x = 5;  
A.y = 6;
```

```
printf("( %d, %d) \n", A.x, A.y); => (5,6)
```

- Nếu dùng con trỏ thì:

```
ptrA->x = 2;  
ptrA->y = 5;
```





# Sử dụng các con trỏ kiểu cấu trúc

- **Có 2 cách:**

- Phải cấp phát bộ nhớ cho nó
- Cho nó chỉ vào (chứa địa chỉ) biến đang tồn tại

- **Ví dụ:**

```
struct NgayThang *p;
```

```
...
```

```
p=(struct NgayThang *)malloc(sizeof(struct NgayThang));  
p->Ngay=29; p->Thang=8; p->Nam=1986;
```

***Hoặc***

```
struct NgayThang Ngay = {29,8,1986};  
p = &Ngay;
```

# Truy cập các thành phần của cấu trúc đang được quản lý bởi con trỏ (1)



- **Với khai báo sau:**

```
struct NgayThang *p;
```

- **Ta có thể truy cập đến các trường của nó như sau:**

`p->Ngay`    `p->Thang`

*hoặc*

`(*p).Ngay`    `(*p).Thang`

# Truy cập các thành phần của cấu trúc đang được quản lý bởi con trỏ (3)



- Ví dụ

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

typedef struct{
    unsigned char Ngay;
    unsigned char Thang;
    unsigned int Nam;
} NgayThang;

int main(){
    clrscr();
    NgayThang Ng={29,8,1986};
    NgayThang *p;
    p=&Ng;
    printf("Truy cap binh thuong %d-%d-%d\n",
        Ng.Ngay, Ng.Thang, Ng.Nam);
    printf("Truy cap qua con tro %d-%d-%d\n",
        p->Ngay, p->Thang, p->Nam);
    printf("Truy cap qua vung nho con tro %d-%d-%d\n",
        (*p).Ngay, (*p).Thang, (*p).Nam);
    getch();
    return 0;
}
```

C:\ TC.EXE

```
Truy cap binh thuong 29-8-1986
Truy cap qua con tro 29-8-1986
Truy cap qua vung nho con tro 29-8-1986
```



Hết chương