



7. Struct

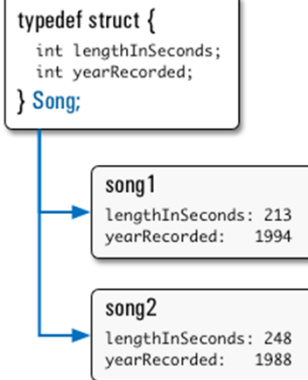
Nội dung

- Khái niệm về Struct
- Khai báo struct
- Truy nhập các trường trong struct
- Phép gán giữa biến kiểu struct
- Con trỏ và struct

3. Cấu trúc

- **Kiểu dữ liệu cấu trúc (struct)** là kiểu dữ liệu phức hợp bao gồm nhiều thành phần, mỗi thành phần có thể thuộc những kiểu dữ liệu khác nhau.

Ví dụ: khi cần lưu giữ thông tin về một dạng đối tượng nào đó như đối tượng sinh viên chẳng hạn, ta lưu giữ các thông tin liên quan đến sinh viên như họ tên, tuổi, kết quả học tập...



3.struct

Khai báo kiểu dữ liệu cấu trúc:

- Để khai báo một kiểu dữ liệu cấu trúc ta dùng cú pháp khai báo sau:

```
struct tên_cấu_trúc
{
    <khai báo các trường dữ liệu>;
};
```

3.struct

```
struct sinh_vien
{
    char masoSV[10];
    char hoten[30];
    float diem_TinDC;
};

struct point_3D
{
    float x;
    float y;
    float z;
};
```

3.struct

Khai báo biến cấu trúc:

- Để khai báo biến cấu trúc ta dùng cú pháp khai báo sau

struct tên_cấu_trúc tên_biến_cấu_trúc;

Ví dụ:

```
struct sinh_vien sv1, sv2;

struct point_3D point1, point2;
```

Câu lệnh trên khai báo 2 biến lần lượt tên là sv1, sv2 có kiểu dữ liệu là cấu trúc **sinh_vien**.

3.struct

- Khai báo dùng từ khóa **typedef**

```
typedef struct point_3D
{
    float x;
    float y;
    float z;
} POINT;
```

Khi khai báo biến chỉ cần dùng **POINT**

POINT x,y; (thay vì **struct point_3D x,y;**)

Xử lí dữ liệu cấu trúc

Truy nhập các trường dữ liệu của cấu trúc :

- Dữ liệu của một biến cấu trúc bao gồm nhiều trường dữ liệu, và các trường này độc lập với nhau.
- Muốn thay đổi nội dung dữ liệu bên trong một biến cấu trúc ta cần truy nhập tới từng trường và thực hiện thao tác cần thiết trên từng trường đó.

tên_biến_cấu_trúc.tên_trường

- Giờ đây ta có thể “đối xử” ***tên_biến_cấu_trúc.tên_trường*** giống như một biến thông thường,

```

struct sinhvien
{
    char masoSV[10];
    char hoten[30];
    float diem;
};

int main(void)
{
    struct sinhvien a;
    strcpy(a.masoSV, "BK2456");
    strcpy(a.hoten, "Nguyen Van A");
    a.diem=7.6;
    printf("Ten : %s MasoSV: %s Diem: %0.2f \n", a.hoten,
        a.masoSV, a.diem);

    return 0;
}

```

3.Struct

Phép gán giữa các biến cấu trúc

- Phép gán cấu trúc có cú pháp tương tự như phép gán thông thường

biến_cấu_trúc_1 = biến_cấu_trúc_2;

- Câu lệnh trên sẽ gán giá trị của các trường trong **biến_cấu_trúc_2** cho các trường tương ứng trong **biến_cấu_trúc_1**.

Một số ví dụ về struct

- Chương trình xác định ngày tiếp theo, với thời điểm ngày nhập vào từ bàn phím.

```
#include <stdio.h>
int main (void)
{
    struct date
    {
        int month;
        int day;
        int year;
    };
    struct date today, tomorrow;
    const int daysPerMonth[12] = { 31, 28, 31, 30, 31, 30, 31, 31,
    30, 31, 30, 31 };
    printf ("Enter today's date (mm dd yyyy): ");
    scanf ("%i%i%i", &today.month, &today.day, &today.year);
```

```
    if ( today.day != daysPerMonth[today.month - 1] ) {
        tomorrow.day = today.day + 1;
        tomorrow.month = today.month;
        tomorrow.year = today.year;
    }
    else if ( today.month == 12 ) { // end of year
        tomorrow.day = 1;
        tomorrow.month = 1;
        tomorrow.year = today.year + 1;
    }
    else { // end of month
        tomorrow.day = 1;
        tomorrow.month = today.month + 1;
        tomorrow.year = today.year;
    }
    printf ("Tomorrow's date is %i/%i/%.2i.\n", tomorrow.month,
        tomorrow.day, tomorrow.year % 100);
    return 0;
}
```


Mảng cấu trúc

- **Mảng cấu trúc** : mỗi phần tử của mảng là một cấu trúc.

```
struct date LichHen[3];
LichHen[1].month = 11;
LichHen[1].day=10;
LichHen[1].year=2010;
```

- Khởi tạo biến mảng cấu trúc

```
struct date lichHen[3]={12,11,2010},{25,11,2010},{5,12,2010}};
struct date lichHen[3]={12,11,2010,25,11,2010,5,12,2010};
```

Mảng cấu trúc

```
int main()
{
    struct sinhvien a[5];
    int i;
    for(i=0; i<5; i++)
    {
        printf("Nhap sinh vien thu %d \n",i+1);
        printf("Ma SV: "); scanf("%s",a[i].masoSV);
        printf("Ho ten: "); fflush(stdin); gets(a[i].hoten);
        printf("Diem: "); scanf("%f",&a[i].diem);
    }
    for(i=0; i<5; i++)
        printf("Ten: %s MaSV: %s diem: %0.2f\n",a[i].hoten,a[i].masoSV,a[i].diem);
    return 0;
}
```

Đổ bộ nhớ đệm bàn phím

Struct chứa struct

```
struct date
{
    int month;
    int day;
    int year;
};
```

```
struct time
{
    int hour;
    int min;
    int sec;
};
```

```
struct dateTime
{
    struct date sDate;
    struct time sTime;
};
```

Struct chứa struct

```
struct dateTime event = {11,6,2010};
event.sTime.hour =8;
event.sTime.min =0;
event.sTime.sec =0;

printf("%.2i/%.2i/%.2i %.2i:%.2i:%.2i",
event.sDate.month,event.sDate.day,event.sDate.year,
event.sTime.hour,event.sTime.min,event.sTime.sec);
```

Struct chứa mảng

```
struct sinhvien
{
    char masoSV[10];
    char hoten[30];
    float diem;
};
```

- Truy cập vào từng phần tử của mảng thành phần như với biến mảng thông thường.

```
struct sinhvien sv;
sv.masoSV[0]='B';
sv.masoSV[1]='K';
sv.masoSV[2]='H';
sv.masoSV[3]='N';
```

Bài tập

- **Bài 1:** Định nghĩa cấu trúc sinh viên (mã sinh viên, họ tên, lớp, điểm trung bình), nhập và in ra danh sách lớp theo cấu trúc sinh viên đó.
 - Tìm và in ra màn hình thông tin về sinh viên có mã sinh viên do người dùng nhập vào từ bàn phím.
 - Tìm và in ra màn hình tên những sinh viên có điểm trung bình lớn hơn 5.

3.Struct

Con trỏ cấu trúc

- Để khai báo một biến con trỏ cấu trúc ta dùng cú pháp khai báo

<tên cấu trúc> * <tên biến con trỏ cấu trúc> ;

- Có 2 cách truy nhập vào trường dữ liệu của cấu trúc từ biến con trỏ cấu trúc là

(*<tên biến con trỏ cấu trúc>).<tên trường dữ liệu>

<tên biến con trỏ cấu trúc>-><tên trường dữ liệu>

3.struct

```
struct sinhvien a,b,*c;
strcpy(a.masoSV,"BK2456");
strcpy(a.hoten,"Nguyen Van A");
a.diem=7.6;

printf("Ten : %s   MasoSV: %s   Diem: %0.2f \n",a.hoten, a.masoSV, a.diem);
b=a;
printf("Ten : %s   MasoSV: %s   Diem: %0.2f \n",b.hoten, b.masoSV, b.diem);
c=(sinhvien*)malloc(sizeof(sinhvien));
*c = b;
printf("Ten : %s   MasoSV: %s   Diem: %0.2f \n",c->hoten, c->masoSV, c->diem);
```