



# 第七章 自定义数据类型





## 本章主要内容

---

7.1 结构体类型、变量定义与访问

7.2 结构体与函数

7.3 链表

7.4 共用体

7.5 枚举类型



## 7.2 结构体与函数

- 结构体变量作为函数形参
- 结构体变量作为函数返回值



## 7.2.1 结构体变量作为函数参数

向函数传递参数包括两种方式:

- 传值
- 传地址



## 7.2.1 结构体变量作为函数参数

### 传值

函数被调用时，系统为形参分配空间，并将实参完整拷贝到形参中，在函数中修改形参的成员时，不会影响外部的实参。

## //分析程序运行过程

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

```
struct Date{
```

```
    int year;
```

```
    int month;
```

```
    int day;
```

```
};
```

```
void printDate(struct Date d){
```

```
    printf(“%d-%d-%d\n”, d.year , d.month , d.day);
```

```
    d.year = 2002;
```

```
    d.month = 11;
```

```
    d.day = 3;
```

```
}
```

```
int main(){
```

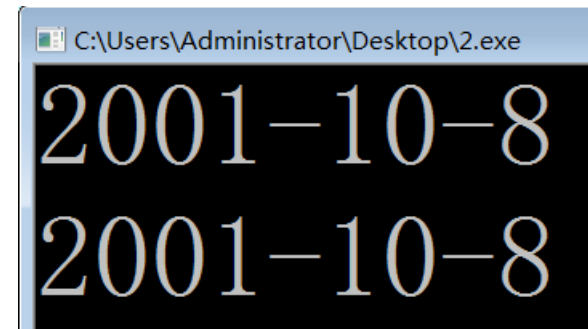
```
    struct Date d = {2001, 10, 8};
```

```
    printDate(d);
```

```
    printf(“%d-%d-%d”, d.year , d.month , d.day );
```

```
    return 0;
```

```
}
```





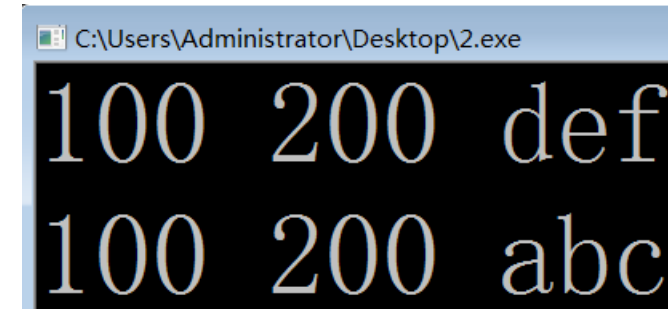
## //分析程序运行过程

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

```
struct sample{
    int a;
    int b;
    char* p;
};

void print(struct sample s){
    printf(“%d %d %s\n”, s.a , s.b , s.p );
    s.a = 300;      s.b = 400;
    strcpy(s.p, "abc");
}

int main(){
    struct sample s1;
    s1.a = 100; s1.b = 200;
    s1.p = new char[20];
    strcpy(s1.p, "def");
    print(s1);
    printf(“%d %d %s\n”, s1.a , s1.b , s1.p );
    return 0;
}
```





## 7.2.1 结构体变量作为函数参数

### 传地址

函数被调用时，将实参地址传递给形参指针，在函数中通过指针修改形参的成员时，会影响外部的实参，因为该指针就是指向的实参数。



## //分析程序运行过程

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

```
struct Date{
```

```
    int year;
```

```
    int month;
```

```
    int day;
```

```
};
```

```
void printDate(struct Date* p){
```

```
    printf(“%d-%d-%d”, p->year, p->month, p->day);
```

```
    p->year = 2002;
```

```
    p->month = 11;
```

```
    p->day = 3;
```

```
}
```

```
int main(){
```

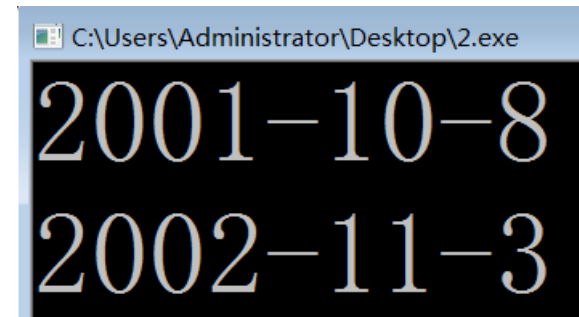
```
    struct Date d = {2001, 10, 8};
```

```
    printDate(&d);
```

```
    printf(“%d-%d-%d”, d.year, d.month , d.day);
```

```
    return 0;
```

```
}
```





## 7.2.1 结构体变量作为函数参数

- **建议**采用传指针方式向函数传递结构体变量，这样可以减少参数传递时的空间，节省处理时间，提高程序效率。
- 如果希望避免在函数出现修改结构体变量成员的语句，可以对形参采用const修饰。如：

```
print(const Date* p)
```



## 7.2.1 结构体变量作为函数返回值

函数可能采用两种方式返回结构体变量：

- 返回一个结构体变量的值
- 返回一个结构体变量的指针



## 7.2.1 结构体变量作为函数返回值 返回一个结构体变量的值

```
#include <stdio.h> //分析程序输出（函数调用及返回过程）
```

```
struct Student{  
    long studentID;  
    char studentName[30];  
    char studentSex;  
    int yearOfBirth;  
    int score[4];  
};
```

```
//根据id号搜索学生信息，并返回该学生的信息
```

```
struct Student search(struct Student* ps, int num, int id){  
    for(int i=0; i<num; i++){  
        if (ps[i].studentID==id){ return ps[i]; }  
    }  
}
```

```
int main(){  
    struct Student stu1 = {100310224,"张明",'M',2001,{90,88,85,89}};  
    struct Student s[50];  
    for(int i=0; i<50; i++){ s[i] = stu1; s[i].studentID = s[i].studentID+i*2;}  
    struct Student s2 = search(s, 10, 100310228);  
    printf("%d", s2.studentID);  
    return 0;  
}
```

如果查找不到给定id的学生呢？  
函数该如何返回？

## //分析程序输出（函数调用及返回过程）

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

```
//struct Student 略 见上页
```

```
//根据id号搜索学生信息，并返回该学生的信息
```

```
struct Student search(struct Student* ps, int num, int id){  
    for(int i=0; i<num; i++){  
        if (ps[i].studentID==id){return ps[i];}  
    }  
    struct Student s={-1};return s;  
}
```

设置一个标志，标识是否查找到指定的学生。学号为-1

```
int main(){  
    struct Student stu1 = {100310224,"张明",'M',2001,{90,88,85,89}};  
    struct Student s[50]; int find  
    for(int i=0; i<50; i++){ s[i] = stu1; s[i].studentID = s[i].studentID+i*2;}  
    struct Student s2 = search(s, 10, 100310228,&find);  
    printf("%d", s2.studentID);  
    return 0;  
}
```



## 7.2.1 结构体变量作为函数返回值 返回结构体变量的指针



**//分析程序输出（函数调用及返回过程）**

**#include <iostream>**

**using namespace std;**

**//struct Student 定义略，见前面**

**//根据id号搜索学生信息，并返回该学生的信息**

**struct Student\* search(struct Student\* ps, int num, int id){**

**for(int i=0; i<num; i++){**

**if (ps[i].studentID==id){ return ps+i; }**

**}**

**return NULL;**

**}**

**int main(){**

**struct Student stu1 = {100310224,"张明",'M',2001,{90,88,85,89}};**

**struct Student s[50];**

**for(int i=0; i<50; i++){ s[i] = stu1; s[i].studentID = s[i].studentID+i\*2;}**

**struct Student\* ps = search(s, 10, 100310228);**

**if (ps!=NULL) printf(“%d”, ps->studentID); else printf(“%s”, “no find”);**

**return 0;**

**}**



河海大学 计算机与信息学院

# 欢迎交流

