

ساختمان ها در زبان C

/Structures In The C

استاد : مهندس محمد سلیمی

گرد آورنده : آتنا شریفی

1389

مقدمه

براساس ساختار آرایه ها ، از مجموعه عناصری تشکیل یافته است که همگی دارای یک نوع داده می باشند . در این مقاله به بحث پیرامون ساختارهایی می پردازیم که عناصر و یا اجزای آنها دارای یک نوع داده های متفاوتی می باشند. در نتیجه یک استراکچر ممکن است دارای اجزاء یا عناصر عدد صحیح ، ممیز شناور و یا از عناصر حرفی تشکیل شود .متغیر های اشاره گر ، آرایه ها و سایر استراکچر ها نیز می توانند به عنوان عنصر در درون یک استراکچر قرار گیرند .

هر یک از اجزای یک استراکچر اصطلاحاً موسوم به عضو و یا **member** می باشد در این مقاله به تعریف و استفاده یک استراکچر و نحوه دستیابی و پردازش هر یک از اعضای آن در برنامه پرداخته شده است .

ارتباط بین استراکچر ها و اشاره گر ها ، آرایه ها و توابع نیز از جمله مباحث مطرح شده در این مقاله می باشد .

۱-۱) تعریف نوع ساختمان

استفاده از استراکچر ها یا همان ساختمان ها به نظر ضروری می رسد . در زبان C نوعی به نام ساختمان وجود ندارد بلکه تمام عناصر ساختمان از انواع موجود ایجاد می شوند . بنابراین قبل از استفاده از ساختمان باید نوع ساختمانی را در برنامه ایجاد و سپس متغیر هایی از آن نوع را تعریف کرد و از آنها استفاده نمود . به عنوان مثال نوع `int` در C وجود دارد برای استفاده از آن نوع باید متغیر هایی را از نوع `int` تعریف کرد و از آنها استفاده نمود . برای تعریف نوع ساختمان از دستور `struct` بهد صورت زیر استفاده می کنیم .

<ساختمان `struct`>

```
{
عناصر ساختمان
};
```

نام ساختمان از قانون نام گذاری برای متغیر ها تبعیت می کند . عناصر ساختمان متغیر هایی هستند که قسمتی از ساختمان می باشند و همانند یک متغیر معمولی و یا آرایه باید اسم و نوع هر کدام مشخص باشد . به مثال زیر توجه کنید .

```
struct personal
{
char name [31];
int person;
int salary;
};
```

در تعریف نوع ساختمان باید موارد زیر را در نظر داشت :

دستور `struct` به ; ختم می شود .

۱-۲) تعریف متغیر نوع ساختمان

با مطالبی که تا کنون گفته شد ، روش تعریف نوع ساختمان مشخص گردید ولی ستعریف ساختمان برای استفاده از آن کافی نیست ، بلکه پس از تعریف نوع ساختمان باید متغیر هایی از این نوع را تعریف کرده سپس این متغیر ها را در برنامه به کار گرفت . تعریف متغیر هایی او نوع ساختمان به دو روش امکان پذیر است .

روش اول : پس از تعریف نوع ساختمان ، متغیر هایی از نوع ساختمان با دستور `Struct` تعریف می شوند . در این روش از شکل کلی زیر استفاده می شود :

; <اسامی متغیرهای ساختمان > نوع ساختمان `struct`

نوع ساختمان ، اسامی است که قبلا در دستور `Struct` ساختار آن مشخص شده است . اسامی متغیر های ساختمان متغیر هایی را مشخص می کند که باید از نوع ساختمانی که اسم آن در نوع ساختمان مشخص شده است تعریف شوند . اگر

تعداد متغیرها بیش از یک باشد با کاما از یکدیگر جدا می شوند . به عنوان مثال دستور زیر دو متغیر ساختمان به اسامی **p1** و **p2** را تعریف می کند که از نوع ساختمان **personel** هستند .

```
struct personal p1,p2;
```

روش دوم : در حین تعریف نوع ساختمان (بلافاصله پس از تعریف نوع) ، اسامی متغیر های ساختمان مشخص می گردند . در این روش از شکل کلی زیر استفاده می شود .

اسم ساختمان **struct**

```
{
```

عناصر ساختمان

```
} ; اسامی متغیر های ساختمان
```

در این روش ذکر اسم ساختمان اجباری نیست . دستور زیر دو متغیر ساختمان به اسامی **p1** و **p2** را تعریف می کند که ساختار آنها توسط دستور **struct** مشخص شده است .

```
struct
```

```
{
```

```
    char name [31];
```

```
    int person ;
```

```
    int salary ;
```

```
} p1 ,p2 ;
```

۳-۱) دسترسی به عناصر ساختمان

همانطور که مشاهده شد هر ساختمان دارای چند عنصر است که برای دسترسی به هر یک از عناصر آن از روش کلی زیر استفاده می شود :

< نام عنصر > . < اسم متغیر از نوع ساختمان >

به عنوان مثال ، با فرض اینکه **p1** و **p2** دو متغیر از نوع ساختمان و دارای ۳ عنصر به اسامی **name** ، **person** و **salary** باشند ، دستورات زیر موجب دسترسی به هر یک از عناصر متغیر **p1** می شوند . در مورد **p2** نیز می توان به همین طریق عمل کرد :

```
p1 . name
```

```
p1 . person
```

```
p1 . salary
```

اگر عناصر ساختمان از نوع آرایه باشند ، ذکر اندیس آرایه جهت دسترسی به عناصر آن الزامی است . به عنوان مثال ، دستور زیر را در نظر می گیریم :

```

struct student
{
    char name[31];
    int cours [10];
    int unit [10];
    int grade [10];
    int person ;
} st1,st2;

```

در این دستور ، بعضی از عناصر ساختمان Student به صورت آرایه تعریف شده اند . دستورات زیر موجب دسترسی به عناصر متغیر ساختمان st1 می شوند .

st1.cours[1]	اولین عنصر آرایه cours
st1.cours[i]	آمین عنصر آرایه cours
st1.name[i]	آمین عنصر آرایه name

همانطور که مشاهده شد ، اگر عناصر ساختمان رشته ای باشند ، می توان به تک تک عناصر آن نیز دسترسی پیدا کرد (مثل st1.name[i]) .

برای مقدار دادن عناصر ساختمان می توان همانند یک متغیر معمولی عمل کرد ، لذا با فرض این که st1 متغیر ساختمانی باشد که قبلا تعریف شده است دستورات زیر معتبر می باشند .

st1.persno =1243;	(مقدار دادن به عنصر ساختمان)
printf (“\n%d”,st1,person);	(چاپ کردن عنصر ساختمان)
gets (st1,name);	(خواندن عنصر رشته ای ساختمان)
for (i=0;i < 10 ; i ++)	(خواندن عنصر آرایه ای ساختمان)
scanf (“ %d “ , &st1 , unit [i]);	

۴-۱) ارزش دهی اولیه به ساختمان

می توان به متغیر های از نوع ساختمان ، همانند آرایه ها مقدار اولیه داد . برای این منظور ، مقادیر اولیه در بین { و } قرار گرفته ، با کاما از یکدیگر جدا می شوند . مقادیری که به متغیر ساختمان نسبت داده می شوند به ترتیب در عناصر متغیر ساختمان قرار می گیرند . اولین مقدار به اولین عنصر ، دومین مقدار به دومین عنصر و n امین عنصر به n امین عنصر نسبت داده می شود . اگر تعداد مقادیر ، از تعداد عناصر ساختمان کمتر باشد ، مقدار بقیه عناصر عددی برابر با صفر و مقدار بقیه عناصر رشته ای برابر با تهی خواهد شد .

نوع ساختمان student را که در زیر آمده است در نظر بگیرید . این دستور علاوه بر تعریف نوع ساختمان دو متغیر s1 و s2 را نیز از نوع student تعریف می کند .

```

struct student
{
    char name[21];
    char address [30];
    int grade [10];
    int stno ;

} st1,st2;

```

مجموعه دستورات زیر را ملاحظه نمایید :

- (1) struct student st1 = { "ali " , " mashhad" , {0} ,15 } ;
- (2) struct student st2 = {0} ;
- (3) struct student st3 = { "ahmad " , '\0' , {15 , 20} , 0 } ;
- (5) struct student st4 = { "reza } ;

در دستور ۱ مقدار ali در name ، mashhad در address صفر در کلیه عناصر آرایه grade و ۱۵ در stno از متغیر st1 قرار میگیرند .

در دستور ۲ کلیه عناصر عددی متغیر st2 مقدار صفر و عناصر کارکتری مقدار تهی را می پذیرند .

در دستور ۳ مقدار ahmad در name ، رشته تهی در address و ۱۵ ، ۲۰ به ترتیب در عناصر اول و دوم آرایه grade و صفر در stno از متغیر st3 قرار می گیرد .

در دستور ۴ مقدار reza در عنصر name از متغیر st4 قرار میگیرد . بقیه عناصر عددی این متغیر مقدار صفر و عناصر غیر عددی مقدار تهی را می پذیرند .

۵-۱) انتساب ساختمان ها به یکدیگر

در گونه های اولیه زبان C انتساب یک ساختمان به ساختمان دیگر امکان پذیر نبود ولی در گونه های جدید C انتساب متغیر های ساختمان که از یک نوع باشند امکان پذیر است . به عنوان مثال اگر st1 و st2 متغیرهایی از نوع ساختمان student باشند . دستور زیر تمام عناصر st1 را به عناصر متناظرش در st2 نسبت می دهد .

```
st2 = st1;
```

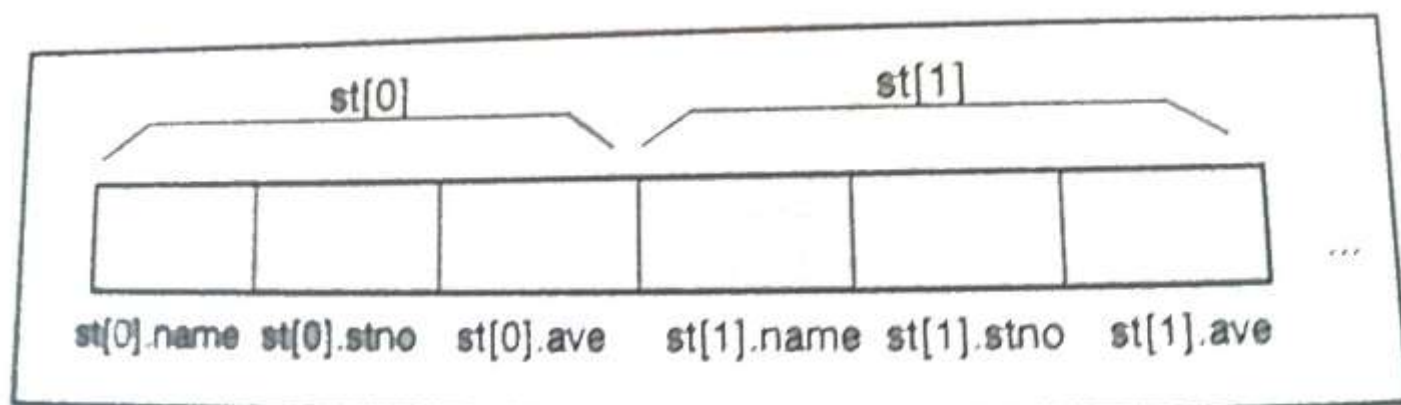
۶-۱) آرایه ای از ساختمان ها

یکی از بیشترین موارد کاربرد ساختمان ها استفاده از آنها به عنوان عناصری از آرایه است برای تعریف آرایه ای از ساختمان ها ابتدا نوع ساختمان را تعریف کرده سپس همانند متغیر های معمولی آرایه ای از آن نوع را تعریف می کنیم . به عنوان مثال دستورات زیر را در نظر بگیرید :

```
struct student
{
    char name[21];
    int stno;
    float ave ;
};

struct student st[100] ;
```

در این دستورات آرایه ۱۰۰ عنصری st طوری تعریف می شود که هر یک از عناصر آن از نوع ساختمان student است شکل زیر را ببینید :



۷-۱) تعریف ساختمان ها به صورت لانه ای

عناصر یک ساختمان می توانند یک متغیر معمولی ، آرایه و یا یک ساختمان باشند . دو مورد اول را قبلا مشاهده کردیم . در این قسمت به بررسی مورد سوم می پردازیم . وقتی که عناصر یک ساختمان از نوع ساختمان باشند برای دسترسی به عناصر آن باید اسامی متغیر های ساختمان را از بیرونی ترین ساختمان به داخلی ترین ساختمان ذکر کرد .

۸-۱) ساختمان ها به عنوان آرگومان تابع

ساختمان ها و یا آرایه ای از ساختمان که تاکنون در برنامه به کار گرفته شده اند یا به صورت عمومی تعریف شدند و یا در تابعی که مورد استفاده قرار می گرفتند تعریف شده اند. ساختمان ها و عناصر آنها می توانند به آرگومان به تابع ارسال شوند.

۹-۱) انتقال عناصر ساختمان به توابع

برای انتقال عناصر ساختمان به تابع ، همانند یک متغیر معمولی عمل می شود با این تفاوت که نام متغیر ساختمان را باید همراه عنصر مورد نظر در آرگومان تابع ذکر کرد . باید توجه داشت که پارامتر تابع نیز باید همنوع با آرگومان متناظر با آن تعریف گردد . به عنوان مثال ساختمان زیر را در نظر بگیرید .

```
struct student
{
    char x;
    int stno;
    char name[31];
}st1 ;
```

هریک از دستورات زیر موجب انتقال مقادیر عناصر متغیر ساختمان st1 به تابعی به نام func() می شوند .

```
func (st1,name [5]);           func (st1,stno);           func (st1,x);
```

برای انتقال آدرس های عناصر متغیر ساختمان به توابع باید از عملگر & استفاده نمود .

۱۰-۱) انتقال ساختمان ها به توابع

متغیر های ساختمان را به دو روش می توان به توابع ارسال کرد ۱ . روش فراخوانی با مقدار ۲ . روش فراخوانی با ارجاع . به طور کلی باید توجه داشته باشید که برای استفاده از متغیر های ساختمان در توابع بهتر است نوع ساختمان در خارج از تابع main() و به صورت عمومی تعریف شود و در داخل توابع ، هر جا نیاز به متغیر های ساختمان بود از آن نوع برای تعریف آنها استفاده گردد .