

Estructuras

martes, 19 de octubre de 2021 07:20 a. m.

Definición (Libro Kernighan "The C programming language"): Es una colección finita de una o más variables, de tipos posiblemente diferentes, agrupadas bajo un solo nombre, para manejo conveniente.

En pocas palabras, una estructura nos sirve para agrupar datos, los cuales tienen una relación entre ellos y de esta forma se facilita el manejo de los datos como un conjunto.

Ejemplo: Un plumón ésta compuesto por una variable llamada volumen de tinta, otra cápsula y otra para color de la tinta.

Sintaxis para creación y manejo de estructuras

Struct declaration--->Definición de estructura:

```
struct Etiqueta
{
    <tipo de dato> identificadorMiembro1;
    <tipo de dato> identificadorMiembro2;
    ...
    <tipo de dato> identificadorMiembroN;
};
```

Nota: La declaración de estructura solamente especifica el nombre y el formato de la estructura, pero no reserva memoria para almacenar los datos.

Struct define (var define)---> Declaración de estructura

Nota: La definición de una variable de tipo estructura, utilizará un área de memoria en donde los datos se almacenan de acuerdo al formato declarado.

```
<struct Etiqueta> identificadorVariableEstructura; //Sintaxis de definición de variable de tipo estructura
<tipo de dato> variable;
```

Instance of struct-> Instancia de una estructura junto a la definición de variables:

```
<struct Etiqueta> identificadorVariableEstructura={<valorParaAtributo1>>, <valorParaAtributo2>>, ...<valorAtributoN>>};
```

Ejemplos:

```
#include <stdio.h>
#define TAM 20
```

```
//variables globales
```

```
//Declaración de estructura
```

```
struct Autor{
    char nombre[TAM]; //20 bytes 0x400-0x419
    int edad; // 4 bytes 0x420-0x423
    double altura; // 8 bytes 0x424-0x431
};
```

```
//Declaración, definir e instanciar de una estructura
```

```
struct Lector{
```

```

    char nombre[TAM];
    int cantidadLibros;
    float presupuestoNuevoLibro;
}lectorJuan={"Juan", 50, 20.30};

```

```

void main(){
    //Definición de una estructura
    struct Autor autorPrincipal, autorSecundario;
    printf("El tamaño de la estructura es: %d\n", sizeof(struct Autor));
    printf("El tamaño de la estructura es: %d\n", sizeof(autorPrincipal));
}

```

Nota: En C es común encontrar operaciones que pueden sustituir a otras, con el fin de mejorar la descripción del programa, su rendimiento o su eficacia. Para el caso de las estructuras, uniones y redefiniciones de tipo de dato, se utilizará la palabra reservada **typedef** para declarar y asignar un nombre alternativo a un tipo de dato; un "alias".

```

#include <stdio.h>
#define TAM 20

//variables globales

//Redefinición de tipo de dato int

typedef int TipoGenerico;

//Declaración de estructura
typedef struct {
    char nombre[TAM]; //20 bytes    0x400-0x419
    TipoGenerico edad; // 4 bytes    0x420-0x423
    double altura;     // 8 bytes    0x424-0x431
} Autor;

//Declaración, definir e instanciar de una estructura

struct Lector{
    char nombre[TAM];
    TipoGenerico cantidadLibros;
    float presupuestoNuevoLibro;
};

//Lector lectorJuan={"Juan", 50, 20.30};

void main(){
    //Definición de una estructura con typedef (ya no se requiere la palabra struct)
    Autor autorPrincipal, autorSecundario;

    //Definición e instancia de una estructura sin typedef( es obligatorio utilizar la palabra struct)
    struct Lector lectorJuan={"Juan", 50, 20.30};

    TipoGenerico miVariableA=70;
    printf("El tamaño de la estructura es: %d\n", sizeof(Autor));
    printf("El tamaño de la estructura es: %d\n", sizeof(autorPrincipal));
    miVariableA++;
    printf("El valor de mi variable es: %d ", miVariableA);
}

```

