

Curso de programación en C moderno (II Edición)

Neira Ayuso, Pablo Falgueras García, Carlos

Tema 4 **Estructuras**

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

1 Struct

- Alineación y tamaño
- Anidamiento
- Estructuras anónimas
- Arrays y punteros

2 Union

- Ejemplo 1
- Ejemplo 2

3 Campos de bits

- Ejemplo

4 Enumerados

- Macros: El preprocesador de C
 - Ejemplos
- Enum
 - Ejemplo

5 Ejercicios: Estructuras

Struct

Portada

Lista de variables agrupadas físicamente en un mismo bloque de memoria.

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

```
1 struct person {
2     char name[256];
3     char surname[256];
4     unsigned char age;
5     unsigned int phone;
6 };
7
8 int main()
9 {
10     struct person p = {"Man", "Bat", 35, 69813244};
11
12     printf("%s%s\n", p.surname, p.name);
13
14     return 0;
15 }
```

Alineación y tamaño

Portada

Struct

Alineación y tamaño

Anidamiento

Anónimas

Arrays y punteros

Union

Ejemplo 1

Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros

Ejemplos

Enum

Ejemplo

Ejercicios

```
struct ejemplo
{
    uint8_t    v1; /* 1 */
    uint32_t   v2; /* 4 */
    uint32_t   v3; /* 4 */
};
```

`sizeof(struct ejemplo);` ¿5 bytes?

Alineación y tamaño

Portada

Struct

Alineación y tamaño

Anidamiento

Anónimas

Arrays y punteros

Union

Ejemplo 1

Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros

Ejemplos

Enum

Ejemplo

Ejercicios

```
struct ejemplo
{
    uint8_t  v1; /* 1 */
    uint32_t v2; /* 4 */
    uint32_t v3; /* 4 */
};
```



`sizeof(struct ejemplo);` ¿5-bytes? → 12 bytes

Anidamiento

Portada

Struct

Alineación y
tamaño

Anidamiento

Anónimas

Arrays y
punteros

Union

Ejemplo 1

Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros

Ejemplos

Enum

Ejemplo

Ejercicios

```
1 struct A {  
2     int a;  
3     struct B {  
4         int b;  
5     } stb;  
6 };  
7  
8 int main()  
9 {  
10     struct A a = {1, {2}};  
11  
12     printf("a = {%d, {%d}}\n", a.a, a.stb.b);  
13  
14     return 0;  
15 }
```

Estructuras anónimas

Portada

Struct

Alineación y
tamaño

Anidamiento

Anónimas

Arrays y
punteros

Union

Ejemplo 1

Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros

Ejemplos

Enum

Ejemplo

Ejercicios

```
1 struct A {  
2     int a;  
3     struct {  
4         int b;  
5     };  
6 };  
7  
8 int main()  
9 {  
10     struct A a = {1, {2}};  
11  
12     printf("a = {%d, {%d}}\n", a.a, a.b);  
13  
14     return 0;  
15 }
```

Arrays y punteros

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas

Arrays y punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

```
1 #include <stdio.h>
2
3 struct cell {
4     int state;
5     int size;
6 };
7
8 int main()
9 {
10     struct cell culture[5] = {{1,10}, {0,342}, {1,7},
11                               {1,50}, {1,77}};
12     struct cell *c;
13     for (c = culture; c < &culture[5]; c++)
14         printf("%d, %d\n", c->state, c->size);
15
16     return 0;
17 };
```


Union

Portada

Struct

Alineación y tamaño
Anidamiento
Anónimas
Arrays y punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

- Una unión es un valor que tiene varias representaciones o formatos
- Estructura que permite guardar varios tipos de datos en la misma zona de memoria

```
1 union float_int {  
2     float f;  
3     int i;  
4 };  
5  
6 int main()  
7 {  
8     union float_int fi;  
9  
10    fi.f = 2.7182;  
11    printf("%X\n", fi.i);  
12  
13    return 0;  
14 }
```

Ejemplo 1

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

```
1 union char_int {
2     struct {
3         char c1;
4         char c2;
5         char c3;
6         char c4;
7     };
8     int i;
9 };
10
11 int main()
12 {
13     union char_int ci;
14
15     ci.i = 1701999205;
16     printf("%c%c%c%c\n", ci.c1, ci.c2, ci.c3, ci.c4);
17
18     return 0;
19 }
```

Ejemplo 2

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

```
1 #include <stdio.h>
2
3 struct gtype {
4     int type;
5
6     union {
7         char char_t;
8         int int_t;
9         float float_t;
10    };
11 };
12
13 int main()
14 {
15     struct gtype gt;
16
17     gt.int_t = 3;
18     gt.type = 1;
19
```

```
20
21     switch (gt.type) {
22     case 0:
23         printf("%c\n", gt.char_t);
24         break;
25     case 1:
26         printf("%d\n", gt.int_t);
27         break;
28     case 2:
29         printf("%f\n", gt.float_t);
30         break;
31     default:
32         printf("error: invalid type\n");
33         break;
34     };
35
36     return 0;
37 }
```

Campos de bits

Portada

Struct

- Alineación y tamaño
- Anidamiento
- Anónimas
- Arrays y punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

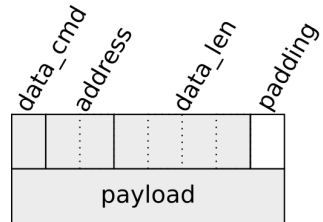
Enumerados

- Macros
- Ejemplos
- Enum
- Ejemplo

Ejercicios

- Característica de las estructuras y uniones que nos permite declarar campos de hasta un bit de longitud
- La memoria reservada es la que indica el tipo del campo
- Para acceder a nivel de bit se realizan numerosas operaciones por debajo

```
struct frame {
    uint16_t data_cmd    : 1;
    uint16_t address     : 2;
    uint16_t data_len    : 4;
    uint16_t             : 1;
    uint16_t payload     : 8;
};
```



Ejemplo

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

```
1 union float_t{
2     float f;
3     struct {
4         uint32_t fra : 23;
5         uint32_t exp : 8;
6         uint32_t sig : 1;
7     };
8 };
9
10 int main()
11 {
12     union float_t f;
13
14     f.f = -3.1416;
15
16     printf("signo      = %u\n", f.sig);
17     printf("exponente = %d\n", f.exp);
18     printf("fraccion  = %d\n", f.fra);
19
20     return 0;
21 }
```

Macros: El preprocesador de C

Portada

Struct

Alineación y tamaño
Anidamiento
Anónimas
Arrays y punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros

Ejemplos
Enum
Ejemplo

Ejercicios

- Preprocesador: Se ejecuta antes de compilar
- Lenguaje de **macros**
- Múltiples usos:
 - Declaración de constantes
 - Pequeñas funciones y utilidades
 - Compilación condicional de código
 - Depuración

Ejemplos

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros

Ejemplos

Enum

Ejemplo

Ejercicios

```
1  #include <stdio.h>
2
3  #define TAM_ARRAY 20
4  #define POW2(x) ((x)*(x))
5  #define PRINT 0
6
7  int main()
8  {
9      int array[TAM_ARRAY];
10     int i;
11
12     for (i = 0; i < TAM_ARRAY; i++)
13         array[i] = POW2(i);
14
15     #if PRINT == 1
16         for (i = 0; i < TAM_ARRAY; i++)
17             printf("%i ", array[i]);
18     #elif PRINT == -1
19         printf("Array initialized\n");
20     #else
21         #warning PRINT may be 1 or -1
22         printf("Error in %s:%d\n", __FILE__, __LINE__);
23     #endif
24
25     return 0;
26 }
```

Enum

Portada

Struct

Alineación y tamaño
Anidamiento
Anónimas
Arrays y punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos

Enum

Ejemplo

Ejercicios

```
enum estado_coche {ARRANCADO, PARADO, EN_MARCHA, DETENIDO};
```

- **Tipo** formado por una lista de macros
- Las macros toman valores enteros de forma consecutiva

Ejemplo

Portada

Struct

Alineación y
tamaño
Anidamiento
Anónimas
Arrays y
punteros

Union

Ejemplo 1
Ejemplo 2

Campos de bits

Ejemplo

Enumerados

Macros
Ejemplos
Enum
Ejemplo

Ejercicios

```

1 #include <stdio.h>
2
3 struct person {
4     char name[256];
5     char surname[256];
6     unsigned char age;
7     unsigned int phone;
8 };
9
10 enum person_attr {
11     NAME,
12     SURNAME,
13     AGE,
14     PHONE
15 };
16
17 int main ()
18 {
19     person p = {"Alice", "Smith",
20                25, 12434321};
21     enum person_attr choice =
22         NAME;

```

```

21
22 switch (choice) {
23     case NAME:
24         printf("%s\n", p.name);
25         break;
26     case SURNAME:
27         printf("%s\n", p.surname);
28         break;
29     case AGE:
30         printf("%d\n", p.age);
31         break;
32     case PHONE:
33         printf("%d\n", p.phone);
34         break;
35     default:
36         printf("error: %s:%d",
37                __FILE__, __LINE__);
38 }
39 return 0;
40 }

```

Ejercicios: Estructuras

Portada

Struct

- Alineación y tamaño
- Anidamiento
- Anónimas
- Arrays y punteros

Union

- Ejemplo 1
- Ejemplo 2

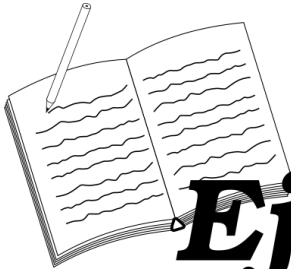
Campos de bits

- Ejemplo

Enumerados

- Macros
- Ejemplos
- Enum
- Ejemplo

Ejercicios



Ejercicios