

Unidad 3

Interfaz del ensamblador con el lenguaje C

SISTEMAS BASADOS EN MICROPROCESADORES

**Grado en Ingeniería Informática
EPS - UAM**

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
-- --
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Índice

Interfaz del ensamblador con el lenguaje C.

1. Características generales.
2. El ejemplo del lenguaje C.
3. Los distintos modelos del lenguaje C.
4. Convenios de nomenclatura, paso de parámetros, devolución de resultados.

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, cursive font. The text is set against a background of a light blue and yellow gradient, with a white speech bubble shape behind it. The logo is positioned in the upper left corner of the slide.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

3.1. Características generales

muchas aplicaciones escritas en lenguajes de alto nivel
quieren partes críticas escritas en ensamblador
ejecución en tiempo real, uso de instrucciones
multimedia tipo MMX, etc.)

permite poder llamar desde programas escritos en
lenguajes de alto nivel compilables a rutinas escritas
en ensamblador.

Además es posible llamar desde programas escritos
en ensamblador a rutinas escritas en lenguajes de alto
nivel compilables.

Esto es posible si programas de ensamblador siguen las
convenciones (nomenclatura, paso de parámetros y
resultados, ...) de los lenguajes de alto nivel.

The logo for Cartagena99, featuring the word 'Cartagena99' in a stylized, green, cursive font. The '99' is larger and more prominent than the 'Cartagena' part. The logo is set against a light blue and white background with a subtle pattern.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

3.2. El ejemplo del lenguaje C (I)

mayoría de aplicaciones que requieren interactuar con un ensamblador escritas en lenguaje C (y C++).

El lenguaje C tiene construcciones típicas de alto nivel (punteros, ciclos, tipos estructurados, recursividad, ...), pero también permite un control a muy bajo nivel (acceso a puertos de E/S, manipulación de bits, ...).

Los compiladores de C permiten el enlace con programas escritos en ensamblador sólo si siguen las mismas convenciones aplicadas por el compilador.

Un programa en C se compila a código objeto, programa ensamblador se ensambla a código objeto y el enlazador (*linker*) del compilador de C genera el ejecutable enlazando esos ficheros objeto.

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, cursive font. The text is positioned on a light blue and white background that resembles a stylized map or architectural drawing.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
...
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

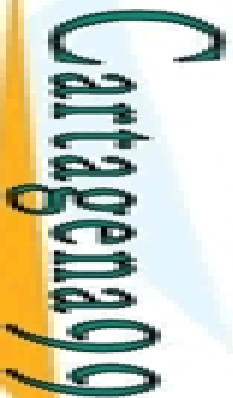
3.2. El ejemplo del lenguaje C (II)

Invenciones del lenguaje C relacionadas con:

Uso de direcciones cortas (*near*) o largas (*far*) para acceder a datos (variables) y/o procedimientos: modelo de memoria.

Nomenclatura de segmentos, variables y procedimientos.

Paso de parámetros a procedimientos y devolución de resultados.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

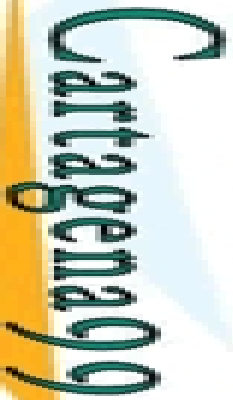
B. Los distintos modelos del lenguaje C (I)

Quando se compila un programa en C se debe escoger un modelo de memoria.

Cada modelo determina la ubicación en memoria de los segmentos lógicos (código, datos y pila) y si se usan direcciones cortas o largas para acceder a ellos.

Los modelos de memoria en Turbo C:

TINY
SMALL
MEDIUM
COMPACT
LARGE
HUGE

The logo for Cartagena99, featuring the word "Cartagena99" in a stylized, green, cursive font. The text is set against a background of a blue and white globe with a yellow and orange swoosh underneath.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

. Los distintos modelos del lenguaje C (II)



Mínima ocupación de memoria.

Los cuatro registros de segmento (**CS**, **SS**, **DS** y **ES**) son idénticos. El programa ocupa hasta 64 KB.

Código, datos y pila en el mismo segmento físico.

Programas compilados en este modelo pueden convertirse a .COM (*drivers*) mediante la utilidad EXE2BIN de DOS o usando la opción /t del montador.

Punteros cortos (*near*) para código y datos.

SMALL

Programas pequeños en los que no es necesario mínima ocupación de memoria.

Un segmento físico para código (hasta 64 KB) y otro para datos y pila (hasta 64 KB).

Punteros cortos (*near*) para código y datos.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Los distintos modelos del lenguaje C (III)

DIUM

Programas grandes que usan pocos datos.

Varios segmentos físicos para código (hasta 1 MB) y uno para datos y pila (hasta 64 KB).

Punteros largos (*far*) para código y cortos (*near*) para datos.

IMPACT

Programas pequeños que usan muchos datos.

Un segmento físico para código (hasta 64 KB) y varios para datos y pila (hasta 1 MB).

Punteros cortos (*near*) para código y largos (*far*) para datos.

LRGE

Programas grandes que usan muchos datos.

Varios segmentos físicos para código (hasta 1 MB) y para datos y pila (hasta 1 MB). En total no puede superarse 1MB.

Punteros largos (*far*) para código y datos.

The logo for Cartagena99, featuring the word 'Cartagena99' in a stylized, green, cursive font. The '99' is larger and more prominent. The logo is set against a light blue and white background with a subtle pattern.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Los distintos modelos del lenguaje C (IV)

IGE

Similar al LARGE con algunas ventajas y desventajas.
Punteros normalizados (offset < 16).
Variables globales estáticas pueden superar 64 KB (posible manipular bloques de datos de más de 64 KB).
Compilador inserta código que actualiza automáticamente registros de segmento de datos (punteros a datos siempre normalizados).
Modelo más costoso en tiempo de ejecución.

The logo for Cartagena99, featuring the word 'Cartagena99' in a stylized, green, cursive font. The '99' is larger and more prominent. The text is set against a background of a stylized cityscape with buildings and a blue sky with a white cloud.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Los distintos modelos del lenguaje C (V)

Modelo	Segmentos			Punteros	
	Código	Datos	Pila	Código	Datos
Modelo 1	64 KB			NEAR	NEAR
Modelo 2	64 KB	64 KB		NEAR	NEAR
Modelo 3	1 MB	64 KB		FAR	NEAR
Modelo 4	64 KB	1 MB		NEAR	FAR
Modelo 5	1 MB	1 MB		FAR	FAR
Modelo 6	1 MB	1 MB (bloques de más de 64 KB)		FAR	FAR



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

 ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP: 689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (I)

Nomenclatura

El compilador de C siempre nombra de igual forma los segmentos lógicos que utiliza:

El segmento de código se llama **_TEXT**.

El segmento **_DATA** contiene las variables globales inicializadas.

El segmento **_BSS** contiene las variables globales NO inicializadas.

El segmento de pila lo define e inicializa el compilador de C en la función *main*.

En los modelos pequeños de datos (*tiny*, *small* y *medium*), todos los segmentos de datos están agrupados con el nombre **DGROUP**:

DGROUP GROUP _DATA, _BSS



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de arámetros, devolución de resultados (II)

Nomenclatura

compilador de C añade un `_` delante de todos los nombres de variables y procedimientos:

Ejemplo:

```
int a = 12345;  
char b = 'A';  
char c[] = "Hola mundo";  
int d = 12;
```

Se compila como:

```
_DATA SEGMENT WORD PUBLIC 'DATA'  
PUBLIC _a, _b, _c, _d  
_a DW 12345  
_b DB 'A'  
_c DB "Hola mundo", 0  
_d DW 12  
_DATA ENDS
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (III)

Nomenclatura

El compilador de C añade un `_` delante de todos los nombres de variables y procedimientos:

Ejemplo:

```
main()
{
    funcion();
}
```

Se compila como:

```
_TEXT SEGMENT BYTE PUBLIC 'CODE'
_main PROC FAR
    CALL _funcion
    RET
_main ENDP
_TEXT ENDS
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (IV)

Nomenclatura

Las variables y procedimientos de ensamblador que sean accedidos desde programas en C deben llevar `_` por delante que no aparece en C.

El lenguaje C distingue entre mayúsculas y minúsculas: `funcion()` y `FUNCION()` son procedimientos distintos.

Es necesario que el ensamblador también distinga entre mayúsculas y minúsculas.

TASM se consigue ensamblando con las opciones `/x` (fuerza distinción para símbolos públicos) o `/ml` (fuerza distinción para todos los símbolos).

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, cursive font. The '99' is larger and more prominent than the 'Cartagena' part. The logo is set against a light blue and white background with a subtle pattern.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (V)

Paso de parámetros

lenguaje C, un procedimiento que llama a otro apila sus parámetros antes de ejecutar el **CALL**.

Los procedimientos de ensamblador que llamen a funciones de C también han de apilar sus parámetros.

Los parámetros se apilan en orden inverso a como aparecen en la llamada de C: empezando por el último acabando por el primero.

Al retornar de la subrutina, se extraen de la pila los parámetros sumando al registro **SP** el tamaño en bytes de los parámetros.

Los parámetros de un byte (char) se apilan con dos bytes (el más significativo vale 0).



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (VI)

Paso de parámetros

Los parámetros se apilan en formato *little endian*:
la palabra menos significativa en dirección menor y byte menos significativo en dirección menor.

Para pasar por parámetro punteros a funciones o a estructuras, es necesario saber en qué modelo de memoria está compilando el programa en C, para apilar el registro de segmento (modelo largo) o no apilarlo (modelo corto).

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, cursive font. The text is positioned above a blue and yellow graphic element that resembles a stylized '9' or a similar shape.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (VII)

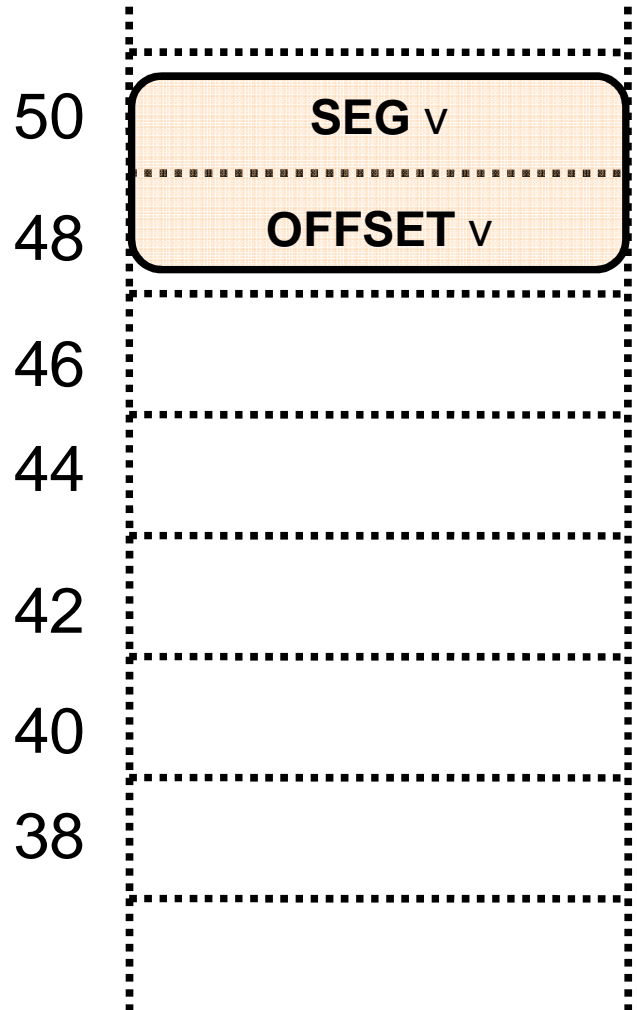
Paso de parámetros (ejemplo)

Prototipo de la función:

void funcion (char, long int, void *);

llamada en modelo largo
enteros **FAR** para datos y
(dígito)

funcion ('P', 0x23, &v);



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
...
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de rámetros, devolución de resultados (VIII)

Paso de parámetros (ejemplo)

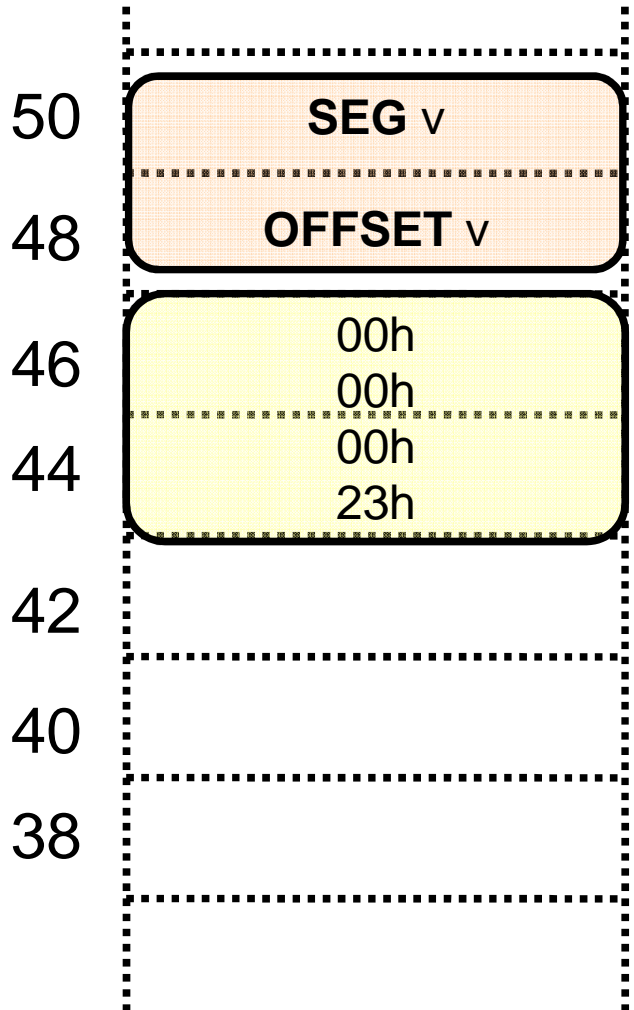
Prototipo de la función:

void funcion (char, long int, void *);

llamada en modelo largo

enteros **FAR** para datos y
(digito)

funcion ('P', 0x23, &v);



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (IX)

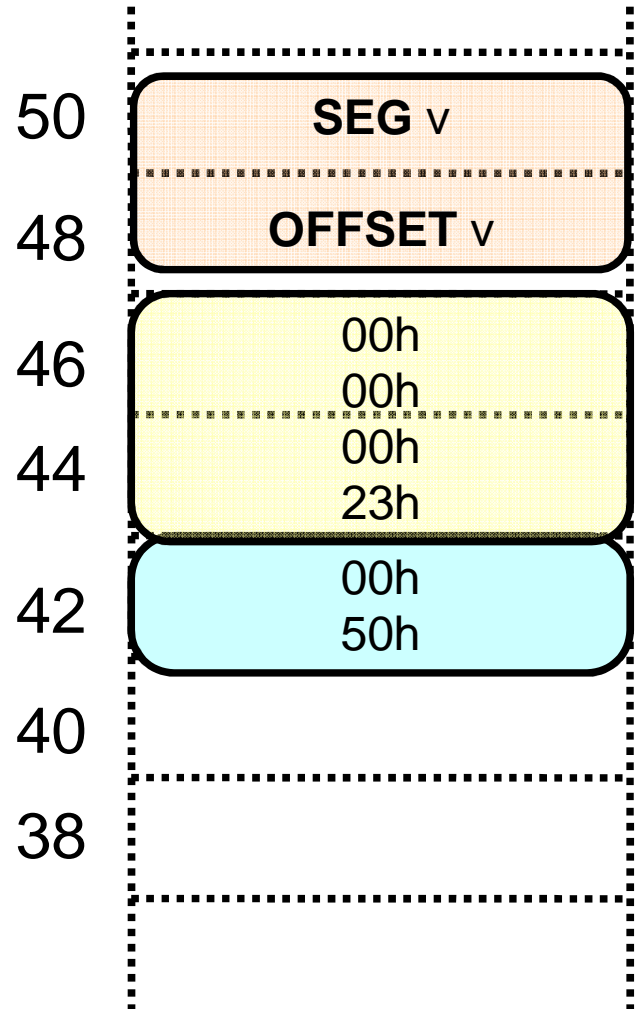
Paso de parámetros (ejemplo)

Prototipo de la función:

void funcion (char, long int, void *);

llamada en modelo largo
enteros **FAR** para datos y
(digito)

funcion ('P', 0x23, &v);



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (X)

Paso de parámetros (ejemplo)

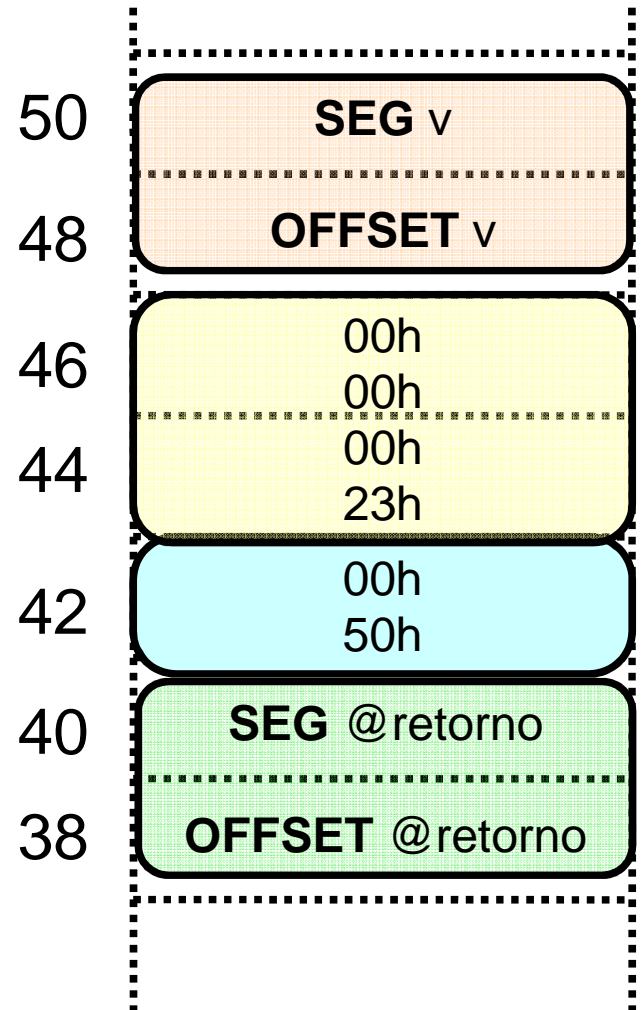
Prototipo de la función:

void funcion (char, long int, void *);

llamada en modelo largo
enteros **FAR** para datos y
(dígito)

funcion ('P', 0x23, &v);

call _funcion



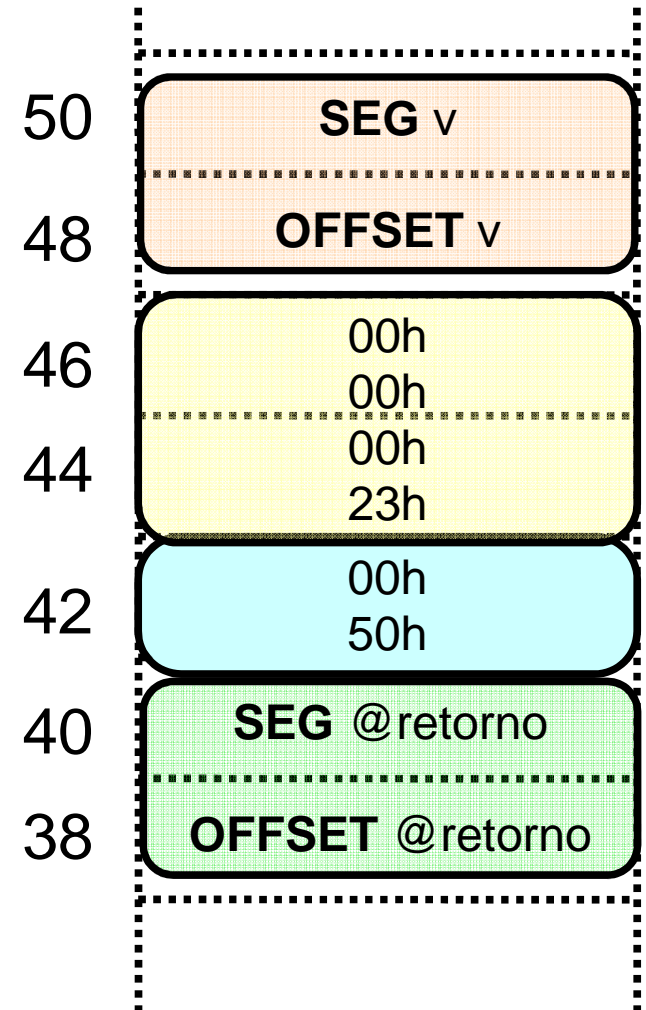
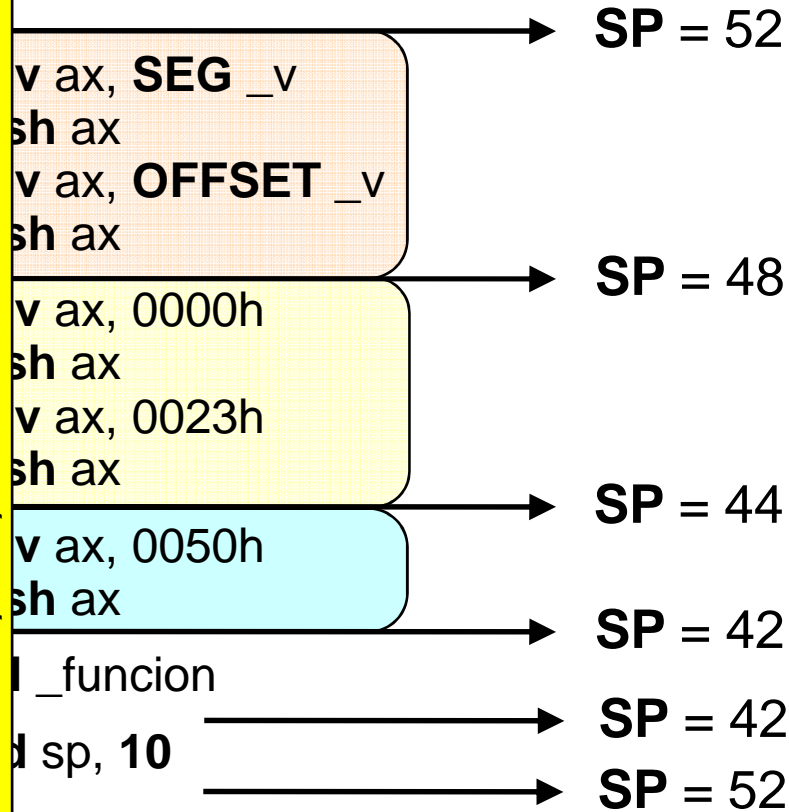
Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (XI)

Paso de parámetros (ejemplo)

id funcion (char, long int, void *);
ncion ('P', 0x23, &v);



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (XII)

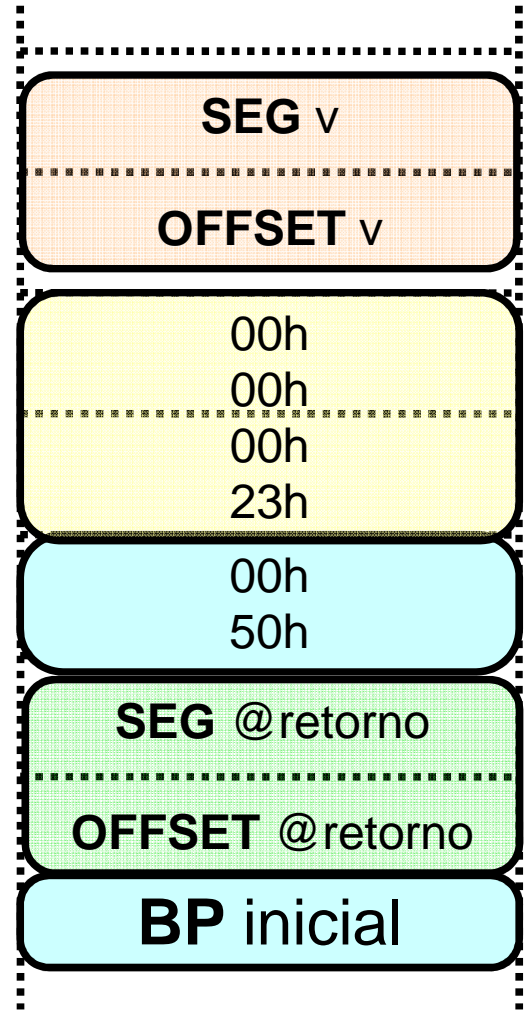
Acceso a parámetros (ejemplo)

funcion (char, long int, void *);

```
on PROC FAR  
push bp  
mov bp, sp  
  
les bx, [ bp + 12 ]  
mov dx, [ bp + 10 ]  
mov ax, [ bp + 8 ]  
mov cx, [ bp + 6 ]  
...  
ret  
on ENDP
```

bx := OFFSET v
es := SEG v

50
48
46
44
42
40
38
36



SP = BP →



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de rámetros, devolución de resultados (XIII)

Devolución de resultados

Las variables de retorno de una función con una longitud de 16 bits se devuelven al procedimiento mediante en **AX** y las de 32 bits en **DX:AX**.

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, cursive font. The text is positioned above a graphic element consisting of a blue and yellow shape that resembles a stylized '9' or a similar character.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

--

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de rámetros, devolución de resultados (XIV)

Ejemplo 1

able visible (externa) desde la rutina de ensamblador */

iable_c;

able definida como pública en ensamblador */

int dato_as;

aración de la función de ensamblador (podría estar en un **include**) */

cion (**int** a, **char** far *p, **char** b) ;

a = 123;

r b = 'F';

r far *p;

/* Declaración de variables locales de C */

na a la función y guarda en variable_c el valor devuelto en AX */

variable_c = funcion (a, p, b);

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, cursive font. The text is set against a background of a blue and yellow gradient with a subtle pattern of small icons.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (XV)



```
JP GROUP _DATA, _BSS ; Se agrupan segmentos de datos en uno

SEGMENT WORD PUBLIC 'DATA' ; Segmento de datos público
PUBLIC _dato_as ; Declaración de _dato_as como público
    _dato_as DW 0 ; Reserva de _dato_as e inicialización
ENDS

SEGMENT WORD PUBLIC 'BSS' ; Segmento de datos público
EXTERN _variable_c : WORD ; Declaración de _variable_c como externa y de
    _variable_c DW ? ; tipo WORD (definida en el módulo C)
    ; Variable no accesible por el módulo C
ENDS

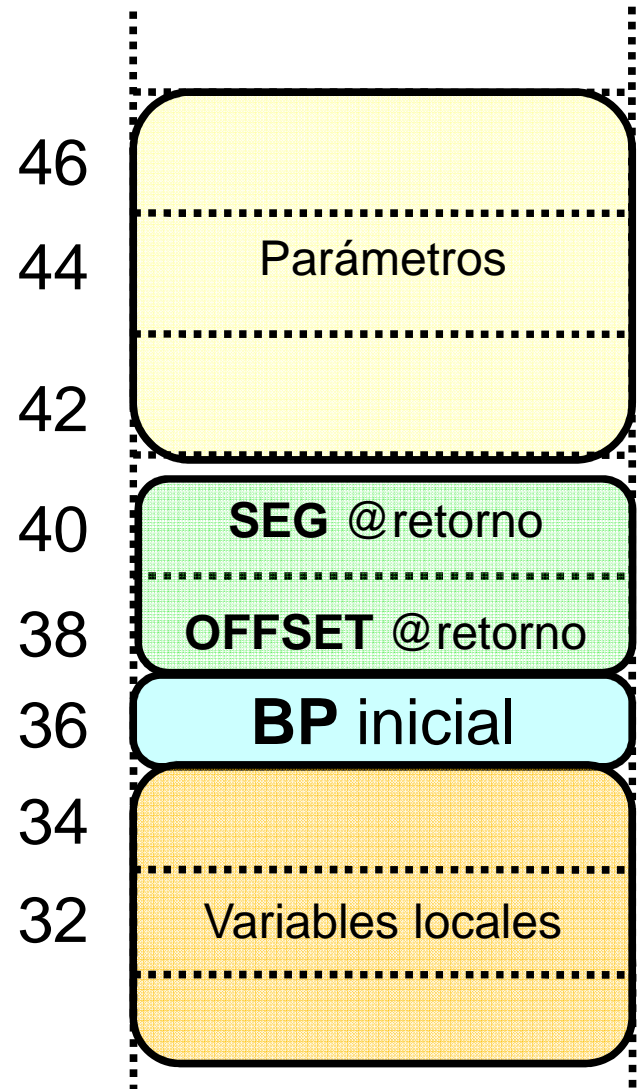
--
SEGMENT BYTE PUBLIC 'CODE' ; Definición del segmento de código
ASSUME CS:_TEXT, DS:DGROUP, SS:DGROUP
C _funcion ; Hace accesible a funcion desde C
C _funcion PROC NEAR ; En C es funcion ()
    PUSH BP ; Para poder utilizar BP para direccionar la pila
    MOV BP, SP ; se carga con puntero a cima de pila
    MOV BX, [ BP+4 ] ; Guarda a en BX , BX=123
    MOV SI, [ BP+6 ] ; Guarda p en DS:SI
    MOV CX, [ BP+10 ] ; Guarda b en CX, CL='F', CH =0
    ;
    MOV AX, CX ; Como función es int el valor se devuelve en AX
    POP BP ; Restaura BP
    RET ; Retorna a procedimiento llamante
ENDP
ENDS
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de parámetros, devolución de resultados (XVII)

Definición de variables locales

Las rutinas de C definen las variables locales que necesitan la pila, por encima de BP. Se introducen en mismo orden que están declaradas. Se acceden mediante [BP-2], [BP-4],



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
...
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4. Convenios de nomenclatura, paso de ámetros, devolución de resultados (XVIII)

instrucción `asm` permite insertar ensamblador en el programa en C (ensamblador *inline*).

```
main
{
    int d1 = 5, d2 = 4, resultado;
    asm {
        push cx
        push ax
        mov ax, 0
        mov cx, d2
        cmp cx, 0
        jz final
    }
    mult:
    asm {
        add ax, d1
        dec cx
        jnz mult
    }
    final:
    mov resultado, ax
    pop ax
    pop cx
}
printf ("resultado %d\n", resultado); }
```

No se debe usar **BP**, ya que lo utiliza el compilador para acceder a las variables locales.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70
--
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70