

# PROGRAMACIÓN EN LENGUAJE C

APUNTADES - PASO DE PARÁMETROS - MEMORIA DINÁMICA

Facultad de Ingeniería  
Elba Karen Sáenz García  
Laura Sandoval Montaña

# APUNTADORES

- Un *apuntador o puntero* es una variable que puede almacenar direcciones de memoria (variables o datos).
- Para declarar una variable apuntador se le antepone al nombre un \* y se le indica el tipo de dato (al que apuntará).

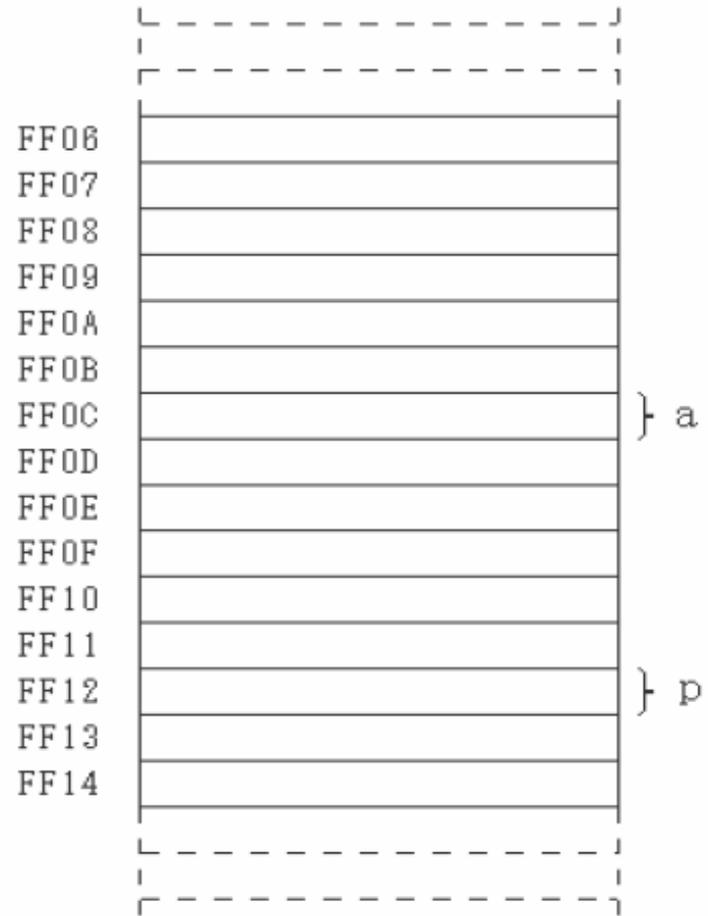
```
char *pa; /* pa es un apuntador a un carácter */
```

```
int *pi; /* pi es un apuntador a un entero */
```

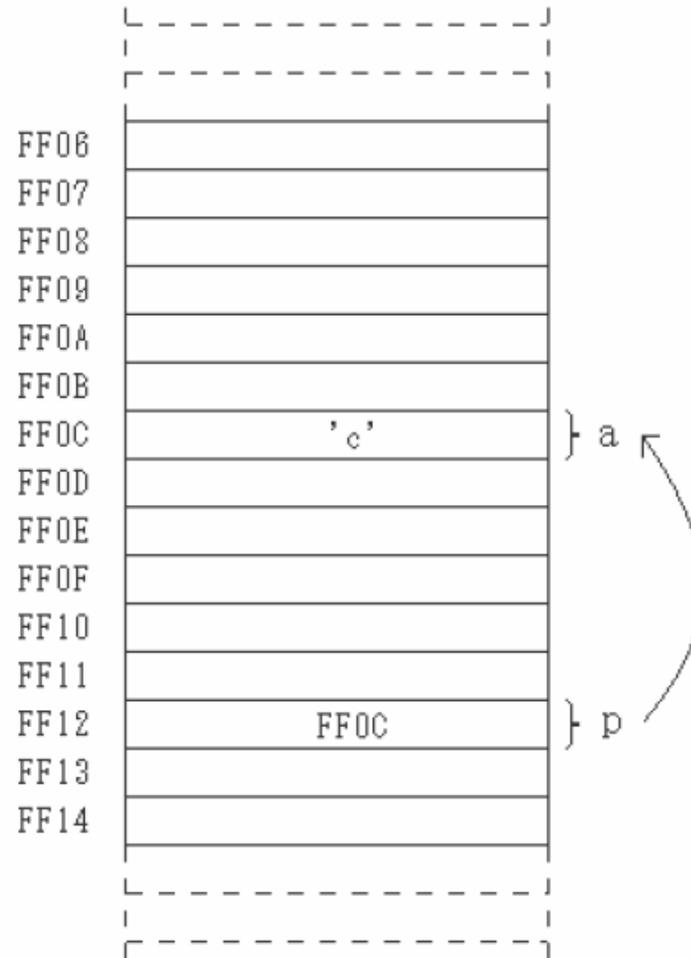
```
float *pf; /* pf es un apuntador a un real */
```

# EJEMPLO

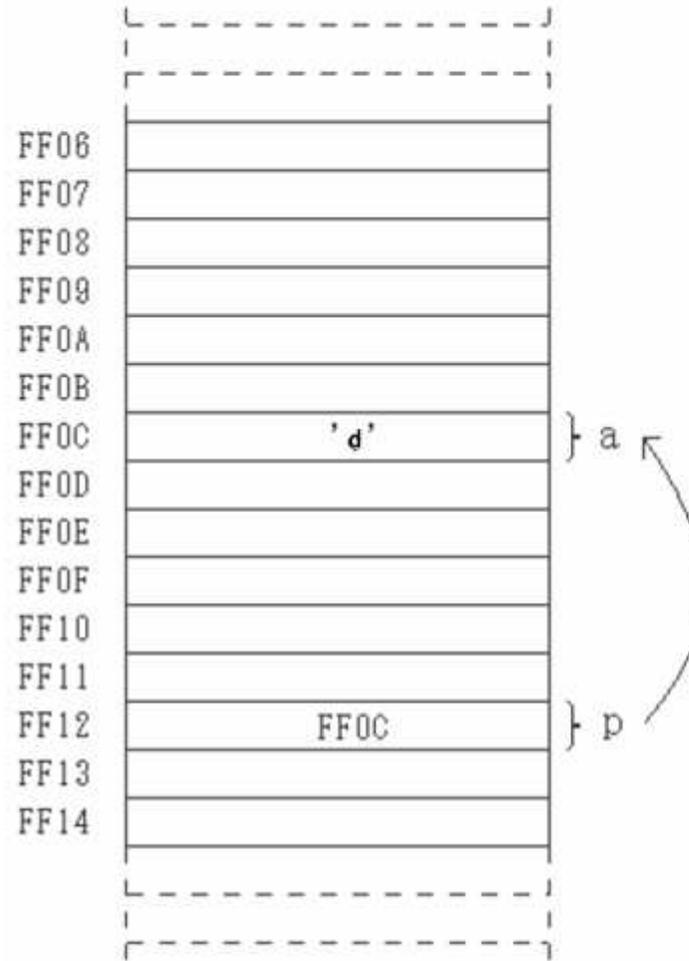
```
char a, *p;
```



```
a = 'c';  
p = &a;
```



`*p = 'd';`



# OPERADORES IMPORTANTES

Dirección (&): Obtiene la dirección de memoria de la variable de la variable a la que precede.

Indirección: (\*) Si se antepone a una variable apuntador se obtiene el valor que contiene la dirección de memoria a la hace referencia el apuntador(a lo que apunta).

# ARITMÉTICA DE APUNTAADORES

Sumar o restar un número entero a un puntero.

$\text{apuntador} + n = \text{Dirección almacenada} + n * \text{tamaño\_tipo\_dato}$

```
char num, *punt, *pnum
```

```
num=1;
```

```
punt = &num;
```

ff00	1	num
ff01		
ff02		
ff03		
ff04		
ff05		
ff06		
ff07		
ff08		pnum
ff09	ff00	punt
ff0a		

# ARITMÉTICA DE APUNTAADORES

Sumar o restar un número entero a un puntero.

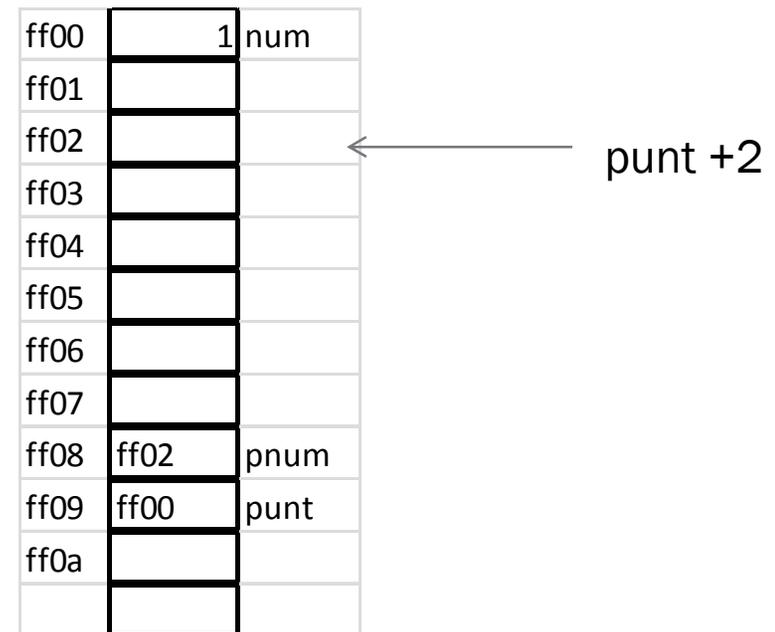
$\text{apuntador} + n = \text{Dirección almacenada} + n * \text{tamaño\_tipo\_dato}$

```
char num, *punt, *pnum
```

```
num=1;
```

```
punt = &num;
```

```
pnum = punt + 2
```



# ARITMÉTICA DE APUNTADES

Sumar o restar un número entero a un puntero.

$\text{apuntador} + n = \text{Dirección almacenada} + n * \text{tamaño\_tipo\_dato}$

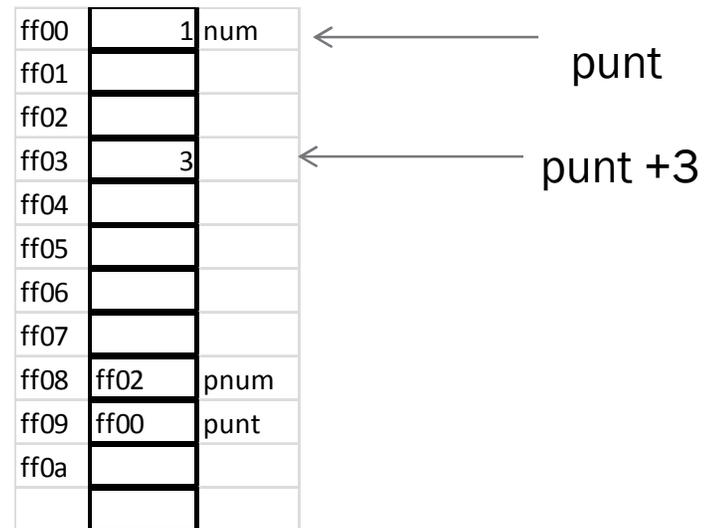
```
char num, *punt, *pnum
```

```
num=1;
```

```
punt = &num;
```

```
pnum = punt + 2
```

```
*(punt+3)=3;
```



# EJEMPLO

```
#include <stdio.h>

main() {
    char valor;
    char *p;
    valor = 97;

    /*Impresión de la dirección con el formato %X, y el contenido de valor*/
    printf("\n %X => | %d | <= dirección y datos de valor. \n", &valor, valor);
    p = &valor;    /*Asignación de valor inicial*/
    /*Impresión de la dirección con el formato %X, y el contenido, con el formato %X, de p*/
    printf(" %X => | %X | <= dirección y datos del apuntador. \n", &p, p);
    printf("\n Valor almacenado en apuntador = %X \n", p);
    printf(" Dirección de apuntador : &p = %X\n", &p);
    printf(" Valor referenciado por apuntador: *p = %d\n", *p);
}
```

# OBSERVACIONES APUNTAORES

*Formatos para el manejo de apuntadores: %x, %X, %p*

*Un apuntador puede asignarse a otro apuntador*

`long a,*p,*q;    p=&a;    p=q;`

## EJEMPLO 2

```
#include<stdio.h>

void main(){
int var,*p,i;

var=0; /*Asignación de valor inicial*/
p=&var; /*Asignación de valor inicial*/

printf("El valor inicial de var es: %d\n",var);

/*Ciclo de lectura y modificación de var*/
for(i=0;i<6;i++){
    printf("nuevo valor: ");
    scanf("%d",p); /*Lectura de valor y asignación a var,usando el apuntador p*/
    printf("var ha cambiado por: %d\n",var);
}
}
```

# RELACIÓN ENTRE UN ARREGLO Y UN APUNTADOR

Relación entre los apuntadores y los arreglos

- Se pueden direccionar arreglos como si fueran apuntadores y apuntadores como si fueran arreglos.

En el lenguaje de programación C, el nombre de un arreglo es el índice a la dirección de inicio del primer elemento del arreglo.

# EJEMPLO

Si se declara un arreglo

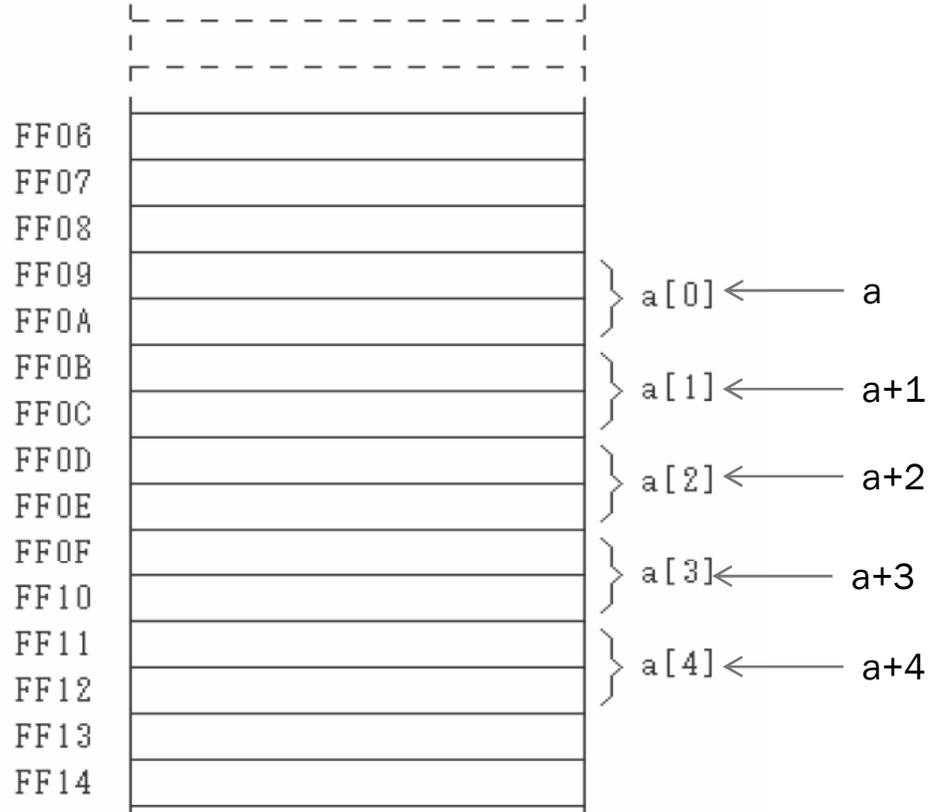
```
int a [5]={2,4,6,8,10};
```

Entonces

a[0] tiene un 2,

a[1] tiene un 4,

a[2] tiene un 6,etc..



a + 0	apunta a	a[0]	que corresponde a la dirección FF09
a + 1	apunta a	a[1]	que corresponde a la dirección FF0B
a + 2	apunta a	a[2]	que corresponde a la dirección FF0D
a + 3	apunta a	a[3]	que corresponde a la dirección FF0F
a + 4	apunta a	a[4]	que corresponde a la dirección FF11

# MANEJO DEL ARREGLO COMO APUNTADOR

- Si se quieren modificar valores

Forma 1

```
*(a+0)=5; *(a+1)=6; *(a+2)=10;*(a+3)=11;*(a+4)=6;
```

Forma 2:

```
for(i=0;i<5; i++)  
    scanf("%d", (a+i));
```

- Imprimir contenido arreglo mediante el apuntador

```
for(i=0;i<5; i++)  
    printf("%d", *(a+i));
```

# EJEMPLO

```
#include<stdio.h>
main() {
int x[5]={1,2,3,4,5};
int *y, renglon;
y = x;
printf("%d %d %d \n",x[1],*y,*(y+1));
for(renglon =0; renglon <= 4; renglon ++)
printf("%d\t",*(y+renglon));
}
```

# MANEJO DE PASO DE PARÁMETROS EN C

- Paso de parámetros por valor: Se hace una copia del valor de los parámetros para trabajarlos en la función. Los parámetros no se modifican.
- Paso de parámetros por referencia: Se utiliza la referencia o dirección para poder modificar los parámetros. O bien, en C se hace una copia de la dirección de memoria de la variable que se quiere modificar

# EJEMPLO

```
void swap(int a, int b);  
main(){  
    int c=2,d=5;  
    swap(c,d);  
}  
void swap(int a, int b){  
    int tmp;  
    tmp=a;  
    a=b;  
    b=tmp;  
}
```

```
void swap(int *a, int *b);  
main(){  
    int c=2,d=5;  
    swap(&c,&d);  
}  
void swap(int *a, int *b){  
    int tmp;  
    tmp=*a;  
    *a=*b;  
    *b=tmp;  
}
```