

Punteros y arreglos

Inicialización

Un puntero puede inicializarse con la dirección de una variable existente. Otra forma de inicializarlo es con NULL. El valor NULL es una constante definida en la librería *stdio.h* que indica que el puntero es nulo, es decir, no apunta a ningún lado aún.

Hay una estrecha relación entre punteros y arreglos que hace que puedan usarse casi de manera indistinta.

El nombre del arreglo es un puntero al primer elemento del mismo. Es decir, si tenemos:

```
int array[5];
```

array equivale a &array[0]

Podríamos cambiar el contenido de la primera posición del arreglo efectuando:

```
int array[5];
```

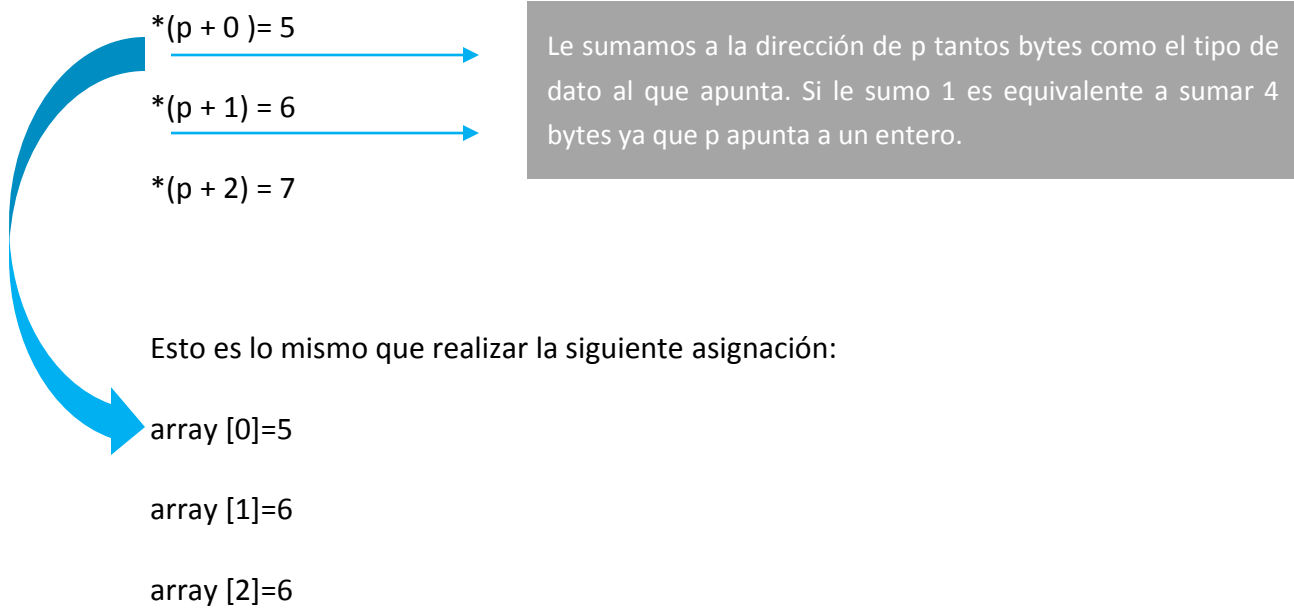
```
int *p=NULL;
```

```
p = array;
```

```
*p = 5;
```

Ya que dicha expresión equivale a `array[0] = 5;`

La aritmética de punteros nos permite acceder a los otros elementos del arreglo:



Al sumar un entero n a un puntero, la dirección almacenada se modifica en n veces el tamaño de lo que apunta. Si el puntero apunta a un char, entonces al sumarle 1 apunta al byte siguiente porque los char ocupan un byte. Pero si el puntero apunta a un entero y el entero es de 4 bytes, deberá sumar 4 bytes a la dirección actual.

¿Cómo se puede mostrar un arreglo utilizando un puntero?

```
for (i=0; i<5; i++)  
{  
    printf("%d\n", *(p+i));  
}
```