

## Punteros en C

Un **puntero** en es una variable que **almacena** como valor **una dirección de memoria**. El tamaño de este tipo de variables es de 8 bytes (en arquitecturas de 64 bits y 4 bytes en arquitecturas de 32 bits).

Será de utilidad almacenar la dirección de alguna variable como valor del puntero, para poder manipular el valor de la variable original.

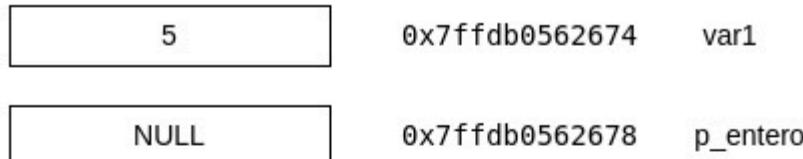
Los punteros pueden ser inicializados con el valor **NULL** o **0**, cuando no se quiere almacenar una dirección en ellos.

```
#include <stdio.h>
int main()
{
    /*
     * Declaramos 2 variables, una de tipo int y
     * una de tipo int* (puntero a entero)
     */
    int var1, *p_entero;

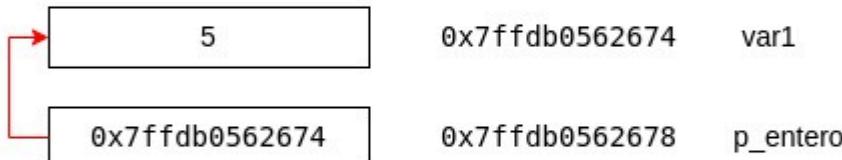
    /* Inicializo variables */
    p_entero = NULL;
    var1 = 5;
    printf("var1: %d \n", var1); /* var1: 5 */
    /* Asigno a p_entero la dirección de memoria de var1 */
    p_entero = &var1;
    /* Modificamos el valor de var1 a través del puntero */
    *p_entero = 20;
    printf("var1: %d \n", var1); /* var1: 20 */
    printf("Valor de var1 desreferenciando p_entero: %d \n", *p_entero);

    /*
     * Podemos conocer las direcciones de memoria involucradas
     * Utilizamos el formateador %p para mostrar direcciones
     * de memoria en notación hexadecimal.
     */
    printf("Dirección de var1: %p\n", &var1);
    printf("Dirección de p_entero: %p\n", &p_entero);
    printf("Dirección de *p_entero: %p\n", p_entero);
    return 0;
}
```

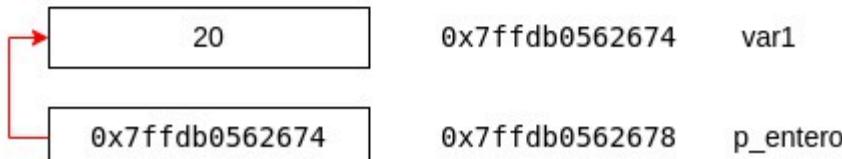
/\* Luego de inicializar las variables \*/



/\* Luego de asignar p\_entero = &var1 \*/



/\* Luego de asignar \*p\_entero = 20 \*/



Podemos obtener la dirección de una variables anteponiendo el carácter & (ampersand). Incluso podemos conocer la dirección donde está alojado el propio puntero.

```
printf("Dirección de var1: %p\n", &var1);
printf("Dirección de p_entero: %p\n", &p_entero);
printf("Dirección almacenada en p_entero: %p\n", p_entero);
printf("Valor de lo apuntado por p_entero: %p\n", *p_entero);
```

Para acceder o modificar el valor de lo apuntado por el puntero, debemos desreferenciar al mismo con la siguiente notación.

```
*p_entero = 20;
printf("Valor de lo apuntado por p_entero: %p\n", *p_entero);
```