



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

Facultad de Ingeniería de Sistemas Computacionales

Capítulo IV

Punteros

Kexy Rodríguez

Contenido

- Conceptos Básicos.
- Declaraciones de Punteros.
- Inicialización.

Conceptos Básicos

“Un puntero es una variable que contiene una dirección de memoria. Normalmente esa dirección es una posición de memoria de otra variable, por lo cual se suele decir que el puntero “apunta” a la otra variable”

Una dirección de memoria permite referenciar rápidamente los elementos de un array. Y en las funciones C facilita modificar los parámetros que recibe.

(Enrique Vicente Bonet Esteban, 2018).

Los punteros se rigen por reglas básicas:

- Un puntero es un variable como cualquiera.
- Una variable puntero contiene una dirección que apunta a otra posición en memoria.
- En esa posición se almacenan los datos a los que apunta el puntero.
- Un puntero apunta a una variable de memoria.

(Janitza de Justiniani, 2012).

Conceptos Básicos

Cuando una **variable se declara**, se asocian tres atributos fundamentales con la misma: su **nombre**, su **tipo** y su **dirección en memoria**.

```
int main() {  
    int x = 20;  
    printf("x = %d", x);  
    printf("\n Dirección en memoria de x = %d", &x);  
  
    return 0;  
}
```

Nombre de la variable	contenido	Dirección
x		1000
		1004
	20	1008
		1012
		1016
		1020
		1024

Operador de dirección: **&** Representa la dirección de memoria de la variable.

Declaraciones de Punteros

Cuando un **puntero se declara**, el tipo base de la declaración sirve para conocer el **tipo de datos** al que pertenece la variable a la cual apunta la variable de tipo puntero. Esto es fundamental para poder leer el valor que almacena la zona de memoria apuntada por la variable puntero y para poder realizar ciertas operaciones aritméticas sobre los mismos.

```
tipo *nombre;
```

El operador ***** aplicado al nombre de un puntero indica el valor de la variable apuntada.

(Enrique Vicente Bonet Esteban, 2018).

Punteros Ejemplo

1

Nombre de la variable	contenido	Dirección
p	?	1000
		1004
x	20	1008
		1012

2

Nombre de la variable	contenido	Dirección
p	1008	1000
		1004
x	20	1008
		1012

```
int main() {
    int x = 20;
    int *p;
    p=&x;
    *p=5;
    printf("\n Dirección del puntero p= %d", p);
    printf("\n Contenido del puntero p= %d", *p);
    printf("\n contenido de la variable x= %d", x);
    return 0;
}
```

3

Nombre de la variable	contenido	Dirección
p	1008	1000
		1004
x	5	1008
		1012

Inicialización de Punteros - NULL

- Normalmente un puntero inicializado adecuadamente apunta a alguna posición específica de la memoria. Sin embargo, un puntero no inicializado, como cualquier variable, tiene un valor aleatorio hasta que se inicializa el puntero. En consecuencia, será preciso asegurarse que las variables puntero utilicen direcciones de memoria válida.
- Un puntero “nulo” no apunta a ninguna parte en particular, es decir un puntero nulo no direcciona ningún dato válido en memoria.
- Un puntero nulo se utiliza para proporcionar a un programa un medio de conocer cuando una variable puntero no direcciona a un dato válido. Para declarar un puntero nulo se utiliza la macro NULL.



UNIVERSIDAD
TECNOLÓGICA
DE PANAMÁ

