

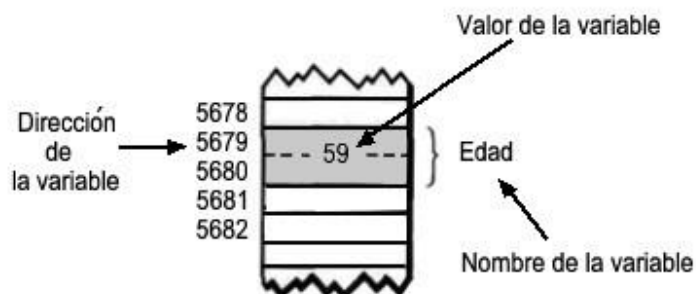
APUNTADORES

Declaración de variables

Antes de abordar el concepto de apuntador es necesario recordar lo que sucede cuando se declara una variable en un programa. Se toma como ejemplo una instrucción en la que se declara una variable entera que almacenará un valor igual a 59:

```
int Edad=59;
```

Con esta instrucción se le comunican 3 cosas al compilador: 1) *el tipo de dato (int)*, 2) *el nombre de la variable (Edad)* y 3) *El valor de la variable (59)*, a lo cual el compilador responde reservando en la memoria el espacio necesario para almacenar dicha información. Podemos ver esto en la siguiente figura:



De esta manera, si después de haber realizado la declaración anterior se escribe la siguiente instrucción:

```
printf("La Edad que ingreso es: %d.", Edad);
```

Al ejecutar el programa se obtendría la salida:

La Edad que ingreso es: 59

Esto es posible ya que cuando se declaró la variable `Edad` se le asociaron 2 valores, el primero es la dirección de memoria en la cual se almacenará la información relacionada con ella y el segundo el valor que se aloja en dicha dirección:

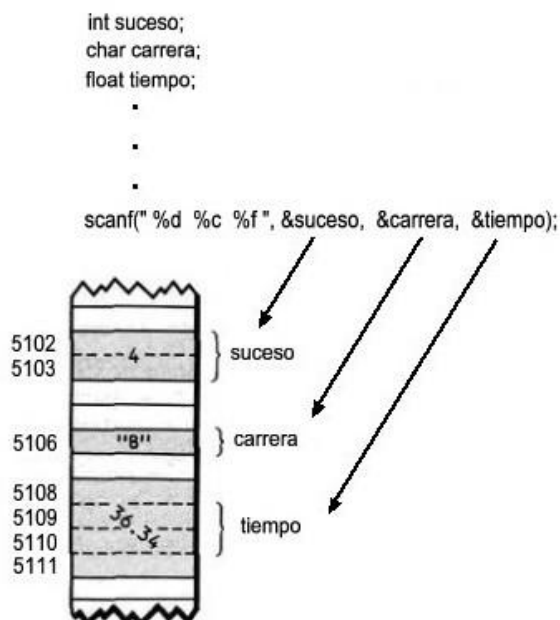
Dirección de la variable `Edad`: 5679.

Valor de la variable `Edad`: 59.

Así, cuando la instrucción `printf` encuentra a la variable `Edad` va a la dirección de memoria reservada para ella (5679), toma el valor almacenado (59) y lo usa para su impresión.

Apuntadores

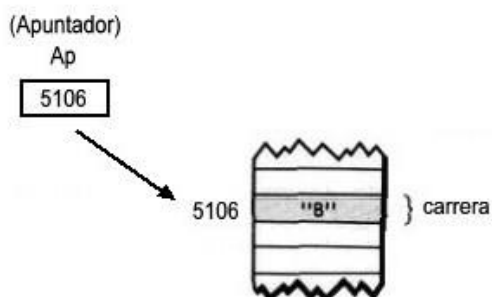
Como se mencionó en los párrafos anteriores cuando se declara una variable el compilador reserva un espacio en la memoria lo suficientemente grande para almacenar el (los) tipo(s) de dato(s) que van a utilizarse. Posteriormente, cada escritura, lectura o actualización del valor de cada variable se realiza en la localidad de memoria asociada, mediante el nombre y dirección de la misma. Como se ejemplifica en la siguiente figura:



Así podemos concluir que una forma de hacer referencia a una variable para almacenar datos, leerlos y/o actualizarlos es mediante su localidad de memoria.

¿Qué es un apuntador?

Para que un programador pueda trabajar con variables mediante su dirección de memoria es necesario el uso de apuntadores. Podemos definir a un apuntador como **una variable que almacena la dirección de otra variable** como se ve en la figura siguiente:



Así, la variable `Ap` es un apuntador cuyo valor es 5106, es decir **apunta** a la dirección 5106, en la cual se almacena la variable `carrera` que tiene un valor de "B".

La sintaxis para declarar un apuntador en lenguaje C es:

Tipo *Ap;

Donde:

- Tipo Es el tipo de la variable apuntada, puede ser int, float, char, etcétera.
- * Es el símbolo que indica que se declara un apuntador.
- Ap Es el nombre de la variable.

Ejemplo: `char Carrera;`
 `char *pCarrera;`

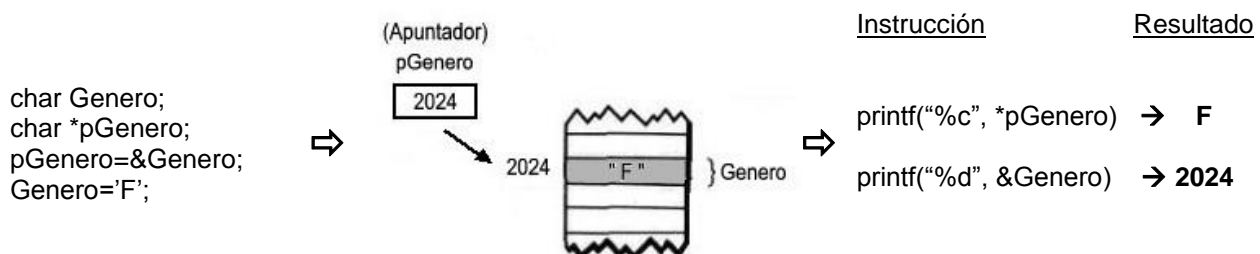
En la primera línea se declara una variable de tipo carácter denominada **Carrera**, en la segunda se declara una variable tipo apuntador llamada **pCarrera** que apuntará a una variable tipo carácter. Se sugiere que cuando se declare una variable apuntador se coloque una **p** después de operador * para indicar que se trata de un apuntador y enseguida se coloque el nombre de la variable a la cual se está apuntando.

Una vez que se ha declarado correctamente la variable apuntador regularmente se realizan operaciones relacionadas con la dirección y el contenido de la variable a la que apuntan. Para lograr esto se hace uso de los operadores **&** y *****:

&: Es un operador unario que devuelve la dirección de memoria del operando.
 (un operador unario es aquel que sólo requiere un operando).

*****: Es un operador unario que regresa el valor que está alojado en la dirección de memoria de la variable que se especifica.

Ejemplo:

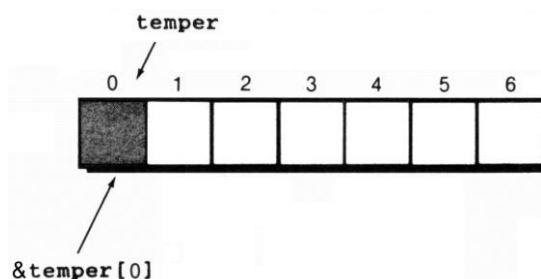


Apuntadores y arreglos

Los conceptos de apuntadores y arreglos tienen una estrecha relación entre ellos, ya que el nombre de un arreglo representa la primera dirección de memoria a dicho arreglo. Por ejemplo si declaramos un arreglo como:

```
Int temper[7]={26, 29, 31, 28, 27, 24, 25};
```

La siguiente condición es verdadera: $temper == \&temper[0]$



Para conocer cómo se pueden usar los apuntadores para recorrer un arreglo hay que recordar la forma como se ha hecho hasta el momento. Por ejemplo para recorrer el arreglo anterior usamos:

```
for(dia=0; dia<7; dia++) /*Se ingresan los datos al arreglo*/  
    printf("\n\t\tEl elemento %d del arreglo es %d.\n\t", dia+1, temper[ i ]);
```

Con apuntadores el acceso a los elementos del arreglo se realiza de forma más rápida ya que se accede a través de su dirección, como se ve en seguida:

```
for(dia=0; dia<7; dia++) /*Se ingresan los datos al arreglo*/  
    printf("\n\t\tEl elemento %d del arreglo es %d.\n\t", dia+1, *(temper+i));
```

Lo mismo es válido para los arreglos de más de una dimensión.