

PRÁCTICA 9
INTRODUCCIÓN A LA PROGRAMACIÓN
(Apuntadores)

Objetivos

1. *El alumno conocerá las características y posibles usos de los apuntadores*
 2. *Conocerá las sintaxis para el manejo de apuntadores*
 3. *Solucionará problemas utilizando apuntadores*
-

¿Qué es un apuntador?

Una variable en un programa es una dirección de memoria reservada que tiene un nombre y que contiene un valor que puede variar. De modo que el compilador y el enlazador (*linker*) asignan un bloque específico de la memoria dentro de la computadora para guardar el valor de una variable. El tamaño de este bloque depende del rango en que a esta variable le es permitido variar. Por ejemplo, en PC's de 32 bits, el tamaño de una variable de tipo entero (int) es de 4 bytes, en una máquina antigua de 16 bits los enteros tienen un tamaño de 2 bytes. Para conocer el tamaño de los diferentes tipos de enteros de tu sistema, ejecutar el siguiente código te dará la información.

```
int main()
{ printf("El tamaño de short (entero corto), es: %d\n", sizeof(short));
  printf("El tamaño de int (entero), es: %d\n", sizeof(int)); printf("El
  tamaño de long (entero largo), es: %d\n", sizeof(long)); }
```

Cuando declaramos una variable le informamos al compilador 2 cosas, el nombre de la variable y el tipo de la variable. Por ejemplo, declaramos una variable de tipo entero llamada k al escribir: `int k`; Cuando el compilador encuentra la palabra “int” de esta instrucción, reserva 4 bytes (en un PC) de memoria para almacenar el valor del entero. También construye una tabla de símbolos. Y en esa tabla agrega el símbolo k y la correspondiente dirección de la memoria en donde esos 4 bytes han sido reservados. Así que si luego escribimos:

```
k = 2;
```

esperamos encontrar al momento de la ejecución, un 2 colocado en el área de memoria reservada para guardar el valor de k. Tiene sentido decir que hay dos valores asociados con

Elaborador por:

Dra. Obdulia Pichardo Lagunas

Dr. Sabino Miranda Jiménez

k, uno es el valor del entero alojado ahí (un 2 en el ejemplo de arriba) y el otro el “valor” de la localidad de la memoria donde se ha guardado, es decir, la dirección de k.

Un apuntador es una variable cuyo valor es la dirección de memoria de otra variable. Se dice que un apuntador “apunta” a la variable cuyo valor se almacena a partir de la dirección de memoria que contiene el apuntador. Por ejemplo, si un apuntador p almacena la dirección de una variable x, se dice que “p apunta a x”.

Referenciación

La referenciación es la obtención de la dirección de una variable. En C y C++ esto se hace a través del operador ‘&’, aplicado a la variable a la cual se desea saber su dirección. Nótese que se trata de un operador unario. Ejemplo:

```
int x = 25;
printf("La dirección de x es: %d\n", &x);
```

Declaración de apuntadores

Para declarar un apuntador se especifica el tipo de dato al que apunta, el operador ‘*’, y el nombre del apuntador. La sintaxis es la siguiente:

```
<tipo>* <identificador>;
```

continuación se muestran varios ejemplos

```
int *ptr; // Apuntador a un dato de tipo entero
char *cad1; // Apuntador a dato de tipo char
float *ptr2; // Apuntador a un dato de tipo float
```

Asignación de apuntadores

Se pueden asignar a un apuntador direcciones de variables a través del operador de referenciación (‘&’) o direcciones almacenadas en otros apuntadores. Ejemplos:

```
int i = 5; int *p,
    *q; p =
    &i;
    // Se le asigna a 'p' la dirección de 'i'
    q
    = p;
    // Se le asigna a 'q' la dirección almacenada en 'p' (la misma de 'i')
```

Elaborador por:

Dra. Obdulia Pichardo Lagunas

Dr. Sabino Miranda Jiménez

Desreferenciación de apuntadores

La desreferenciación es la obtención del valor almacenado en el espacio de memoria donde apunta un apuntador. En C esto se hace a través del operador '*', aplicado al apuntador que contiene la dirección del valor.

Ejemplos:

```
int x = 17,  
y; int *p; p  
= &x;  
printf("El valor de x es: %d \n", *p);  
// Imprime 17  
y = *p + 3;  
// A 'y' se le asigna 20
```

Apuntadores para paso de parámetros por referencia

El lenguaje C no provee una manera de pasar parámetros por referencia. Sin embargo, es posible hacerlo a través del uso de apuntadores.

```
#include <stdio.h>  
#include <conio.h>  
  
int suma(int *a, int *b)  
{  
    int r;  
    r = *a + *b;  
    return r;  
} int  
main()  
{  
    int x;    int a=5, b=7;    x  
= suma(&a,&b);    printf( "7  
+ 5 = %d \n", x);    getch();  
    return 0;  
}
```

Elaborador por:

Dra. Obdulia Pichardo Lagunas

Dr. Sabino Miranda Jiménez

Ejercicios propuestos

1) Realice un programa que declare un arreglo de enteros con 5 localidades y solicite los datos al usuario. Una vez obtenidos los datos, el arreglo tiene que ser enviado a una función (utilizando apuntadores) que mostrará en pantalla el dato acompañado de su dirección de memoria.

2) Cree una función tipo void que intercambie el valor de los dos números enteros que se le indiquen como parámetro, los números ya intercambiados se imprimirán en la función main(). Para ello se pasarán como parámetros las direcciones de las variables.

3) Con ayuda de un menú elabore un programa con cuatro funciones tipo void donde calcule:

a) Suma.

b) Resta.

c) Multiplicación.

d) División.

Cada función tendrá tres parámetros, dos por valor (los dos números que recibe) y un parámetro por referencia (el resultado), ya que este último se calculará dentro de la función respectiva. Los datos de entrada y las impresiones de los resultados respectivos se llevarán a cabo en la función main().

Referencias

Programación en C++, Luis Joyanes Aguilar, editorial Mc Graw Hill (2000)

Tutorial sobre apuntadores y arreglos en C, Ted Jensen, Versión 1.2 Febrero de 2000

Elaborador por:

Dra. Obdulia Pichardo Lagunas

Dr. Sabino Miranda Jiménez