

Práctica 8 de C: 10-03-2009

Objetivos:

- Profundizar en los conceptos de punteros y operaciones con punteros en C.
- Asentar los conocimientos sobre definición, manejo y paso de tipos estructurados a funciones en C.

Enunciado:

- Descarga el programa Ejemplo de Asignación de Tipos Estructurados en C de la página web. Analiza su contenido, fijándote sobre todo en:
 - ¿Cómo se accede a los campos de una estructura y a sus direcciones de memoria?
 - ¿Cómo se asignan valores entre variables de tipo estructurado?
- Modifica el programa anterior, añadiendo un algoritmo que me permita mostrar el contenido de la variable estructurada:

Void muestra(direccion d)

Realiza una llamada para mostrar el contenido de las variables *uno* y *dos* utilizando la función *muestra()*.

- Cuando manejamos tipos estructurados no se pueden hacer todas las operaciones básicas (aritmético-lógicas) sobre las variables de tipo estructura. Intenta mostrar el resultado de comparar *uno* con *dos* (`uno == dos`). ¿Qué ocurre?
- Añade una función *iguales()* que compare el valor de ambas y me diga si son iguales o no (se entiende que son iguales si todos sus campos son iguales):

int iguales(direccion d1, direccion d2)

indicará si *d1* y *d2* contienen la misma información (1) o no (0). Pruébalo en el programa principal.

- Añade en último lugar una función que lea los datos de un tipo estructurado. Debe respetar las siguientes especificaciones:

Algoritmo lee_direccion (d)es

D: dirección; {parámetro dato-resultado}

La función en C no debe devolver la estructura *d*. Sólo debe pasarla como parámetro por referencia.

En C existe un operador (`var->campo`) que equivale a la notación `(*var).campo`. Modifica ahora la función *lee_direccion()* para que utilice la notación `var->campo`.

Pruéba ambas ideas en el programa principal.