

Apuntadores y Arreglos

Apuntadores y Argumentos de Funciones

En C, los argumentos de una función pueden pasarse por valor o por referencia. Cuando se pasan por valor, la función recibe una copia de la variable, lo que significa que cualquier modificación dentro de la función no afecta la variable original. Sin embargo, cuando se usan apuntadores, se pasa la dirección de la variable, permitiendo que la función modifique directamente su contenido.

Ventajas del uso de apuntadores en funciones

- Permiten modificar variables fuera del ámbito de la función.
- Optimizan el uso de memoria al evitar copias innecesarias de datos.
- Son esenciales para trabajar con estructuras de datos dinámicas.

Ejemplo

```
#include <stdio.h>

void incrementar(int *ptr) {
    (*ptr)++; // Incrementa el valor al que apunta ptr
}

int main() {
    int n = 5;
    printf("Primero vale: %d\n", n);
    incrementar(&n);
    printf("Ahora Vale: %d\n", n);
    return 0;
}
```

Apuntadores y Arreglos

En C, un arreglo es esencialmente un apuntador al primer elemento de la secuencia. Esto significa que se pueden manipular sus elementos utilizando notación de apuntadores en lugar de índices.

Aspectos a Considerar

- `arr[i]` es equivalente a `*(arr + i)`.
- La dirección base de un arreglo `arr` es `&arr[0]`.
- Un apuntador puede recorrer un arreglo usando aritmética de apuntadores.

Ejemplo

```
#include <stdio.h>

int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int *ptr = arr; // Apunta al primer elemento

    for (int i = 0; i < 5; i++) {
        printf("Elemento %d: %d\n", i, *(ptr + i));
    }
    return 0;
}
```

Aritmética de Direcciones

La aritmética de apuntadores permite desplazarse por la memoria de manera eficiente. En C, se pueden realizar las siguientes operaciones:

- Incrementar ($\text{ptr}++$): Avanza al siguiente elemento del tipo de dato apuntado.
- Decrementar ($\text{ptr}--$): Retrocede al elemento anterior.
- Sumar ($\text{ptr} + n$): Desplaza el apuntador n posiciones hacia adelante.
- Restar ($\text{ptr} - n$): Desplaza el apuntador n posiciones hacia atrás.

Ejemplo

```
#include <stdio.h>
```

```
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int *ptr = arr;  
  
    for (int i = 0; i < 5; i++) {  
        printf("Valor: %d, Dirección: %p\n", *(ptr + i), (ptr + i));  
    }  
    return 0;  
}
```

Apuntadores a Caracteres y Funciones

Las cadenas en C se representan como arreglos de caracteres terminados en `\0`. Se pueden manipular utilizando apuntadores, facilitando operaciones como concatenación, copia y comparación.

Ejemplos

```
#include <stdio.h>

void imprimirCadena(char *cadena) {
    while (*cadena) {
        printf("%c", *cadena);
        cadena++;
    }
    printf("\n");
}

int main() {
    char texto[] = "Hola, mundo";
    imprimirCadena(texto);
    return 0;
}
```

Referencias

Deitel, P., & Deitel, H. (2015). *C How to Program* (8th ed.). Pearson.

Balagurusamy, E. (2019). *Programming In Ansi C*. McGraw Hill. India