

Programación Avanzada (TC2025)

Tema 1. Programación en lenguaje C

Tecnológico de Monterrey. Campus Santa Fe

Departamento de Computación

Dr. Vicente Cubells (vcubells@tec.mx)

Temario

- Apuntadores y las llamadas por referencia
- Apuntadores a funciones

Apuntadores y las llamadas a funciones por referencia

- Utilizar llamadas por valor siempre que sea posible
- Las estructuras deben pasarse por referencia, de manera explícita
- Utilizar el calificador **const**
 - Reforzar el principio de menor privilegio
 - Existen 6 posibilidades de utilizar o no **const** con parámetros de funciones
 - 2 para paso por valor
 - 4 para paso por referencia

Apuntadores y las llamadas a funciones por referencia

- Utilizar el calificador **const** para pasar parámetros a las funciones
 - Un apuntador *no constante* a un dato *no constante*
 - Ejemplo: función que convierte una cadena a mayúsculas
 - Un apuntador *constante* a un dato *no constante*
 - Ejemplo: el nombre de un arreglo
 - Un apuntador *no constante* a un dato *constante*
 - Ejemplo: función que imprime los caracteres de una cadena
 - Un apuntador *constante* a un dato *constante*
 - Ejemplo: pasar a una función, un arreglo que no se puede modificar

Apuntadores a funciones

- Almacenan la dirección de memoria donde comienza el código de una función
- Pueden:
 - pasarse a una función
 - devolverse desde una función
 - almacenarse en arreglos
 - asignarse a otros apuntadores
- Ejemplo de ordenamiento ascendente y descendente
- Ejemplo del uso de apuntadores a funciones para crear un sistema basado en menús

Resumiendo

- El uso de **const** es fundamental para garantizar el principio de menor privilegio
- Los punteros a funciones permiten crear funciones más versátiles