

# *Apuntadores en Lenguaje C*

## **Programación Digital I**

**Gilberto Diaz**  
**gilberto@ula.ve**  
**Universidad de Los Andes**  
**Facultad de Ingeniería**  
**Escuela de Sistemas**  
**Depto de Computación**  
**Mérida - Venezuela**

# *Apuntadores en Lenguaje C*

- ➔ Los apuntadores son una parte fundamental de C.
- ➔ Si usted no puede usar los apuntadores apropiadamente entonces esta perdiendo la potencia y la flexibilidad que C ofrece básicamente.
- ➔ El secreto para C esta en el uso de apuntadores.

# *Apuntadores en Lenguaje C*

- ➔ C usa los apuntadores en forma extensiva. ¿Porqué?
- ➔ Es la única forma de expresar algunos cálculos.
- ➔ Se genera código compacto y eficiente.
- ➔ Es una herramienta muy poderosa.

# *Apuntadores en Lenguaje C*

C usa apuntadores explícitamente con:

- ➔ Arreglos,
- ➔ Estructuras y
- ➔ Funciones

# ***Definición de Apuntador***

Un apuntador es una variable que contiene la dirección en memoria de otra variable. Se pueden tener apuntadores a cualquier tipo de variable.

# *Definición de Apuntador*

El operador unario & devuelve la dirección de memoria de una variable.

El operador de indirección o dereferencia \* devuelve el ``contenido de un objeto apuntado por un apuntador".

# ***Declaración de Apuntador***

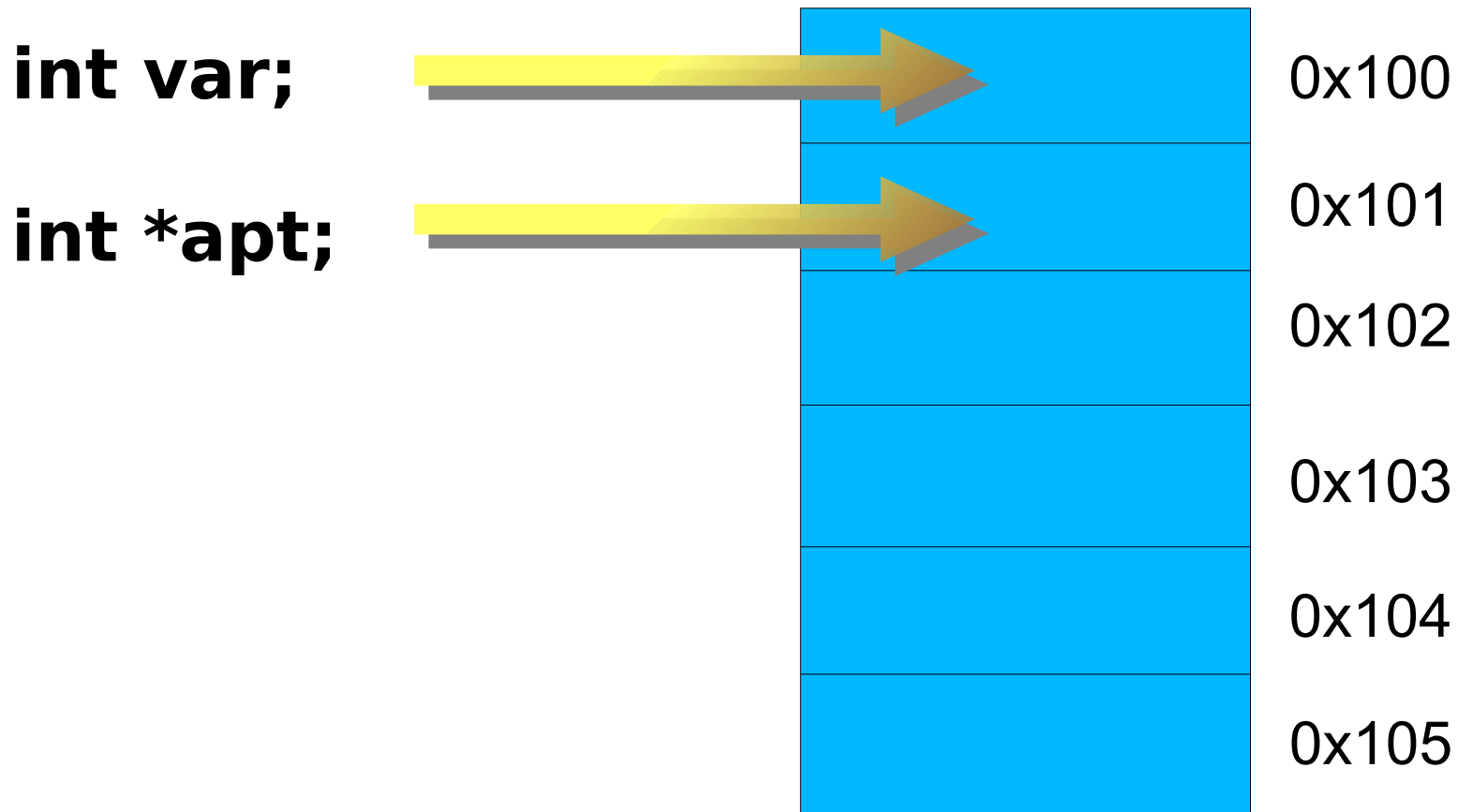
Para declarar un apuntador para una variable entera hacer:

**int \*apuntador;**

Se debe asociar a cada apuntador un tipo particular. Por ejemplo, no se puede asignar la dirección de un short int a un long int.

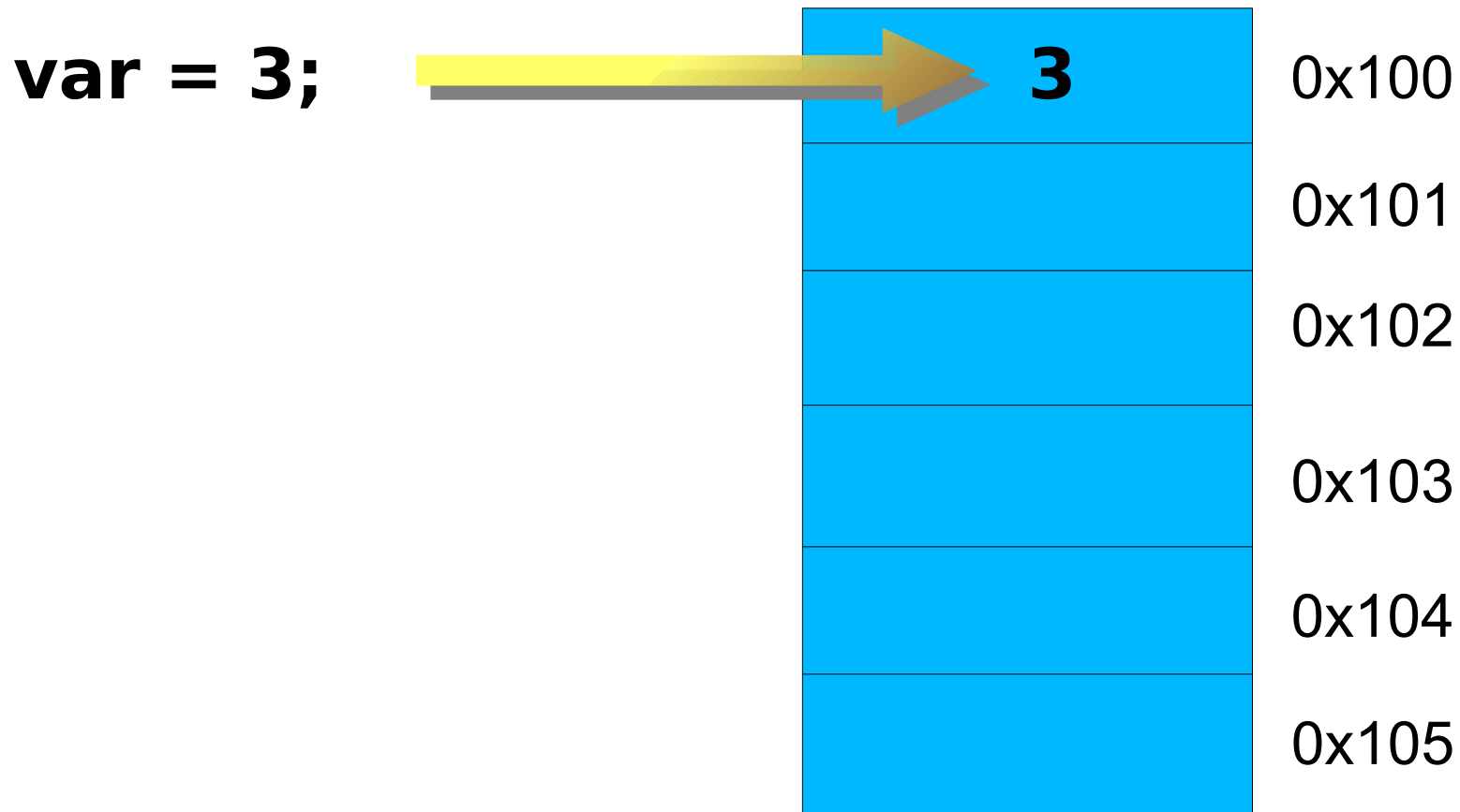
# *Declaración de Apuntador*

Gráficamente tenemos:



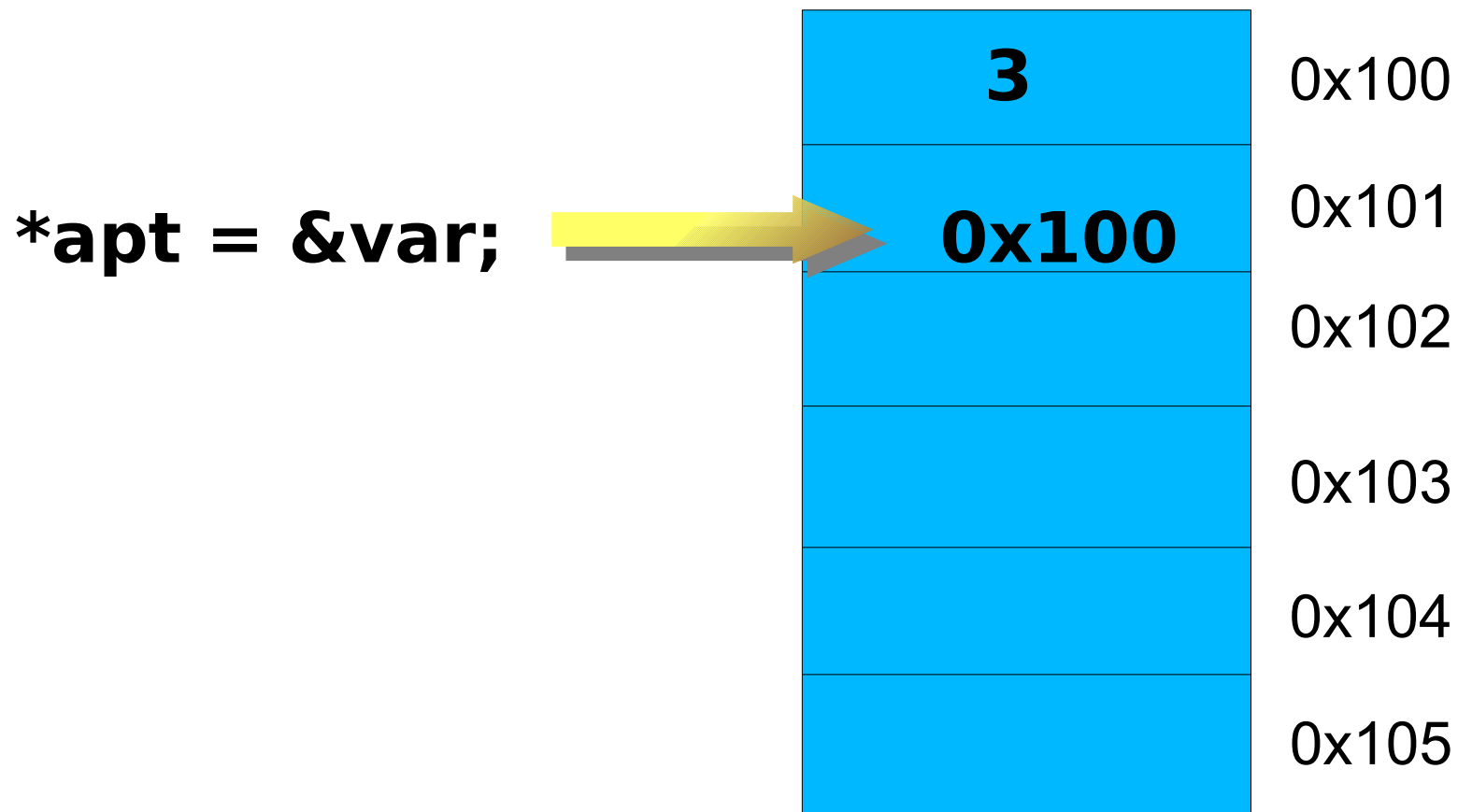
# *Declaración de Apuntador*

Gráficamente tenemos:



# Declaración de Apuntador

Gráficamente tenemos:



# *Parámetros de Funciones*

**Por valor:** Se le pasa a la función una copia del valor (Se crea otra variable con el mismo valor)

**Por referencia:** Se le pasa la dirección de memoria donde se encuentra el valor original