

INTRODUCCIÓN AL MANEJO DE MEMORIA





Memoria RAM

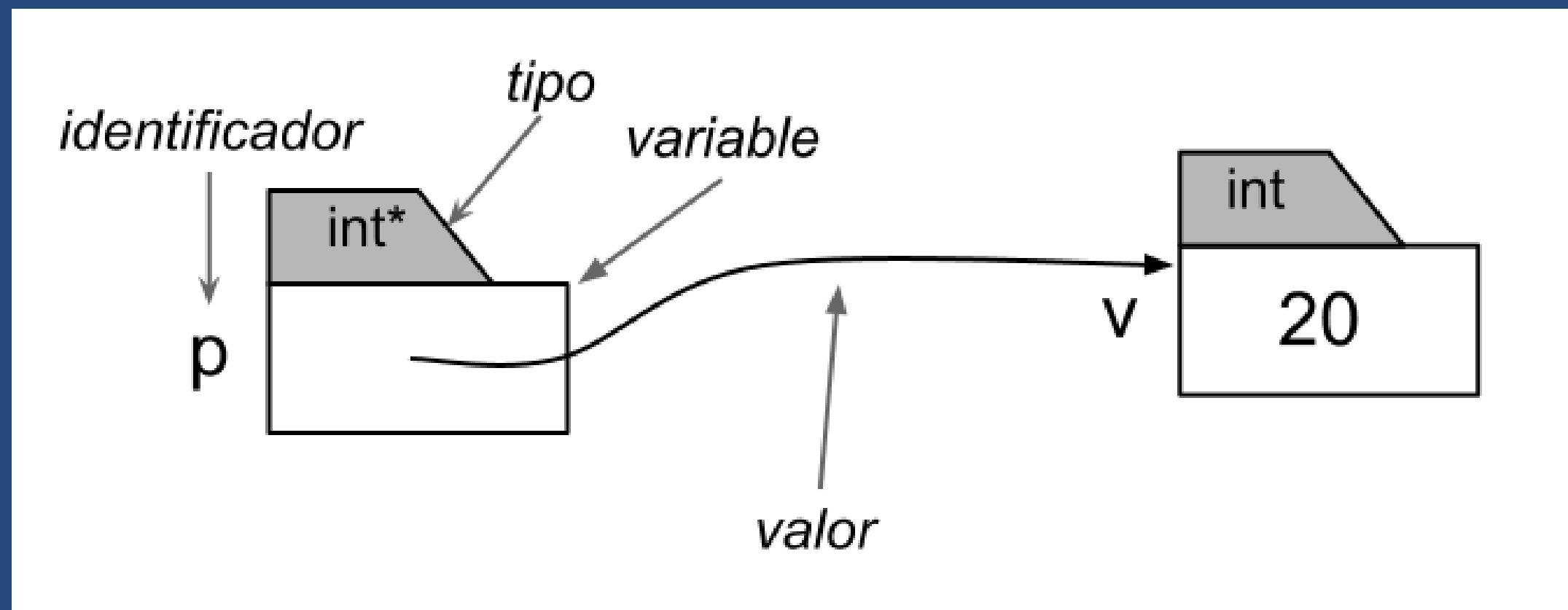
- Espacio de ejecución de nuestros programas hasta el momento (Stack + Heap).
- Es volátil.
- —————
- Secuencia lineal de celdas (bloques).
- Cada celda tiene un identificador único (Su dirección física).
- Una variable es solo una “etiqueta” humana para esa dirección.

**Repaso: ¿Qué
es un puntero?**



Repaso: ¿Qué es un puntero?

- Es una variable.
- Almacena direcciones de memoria.



IMPRIMIENDO PUNTEROS



Imprimiendo_punteros.c



IMPRIMIENDO PUNTEROS



Imprimiendo_punteros.c

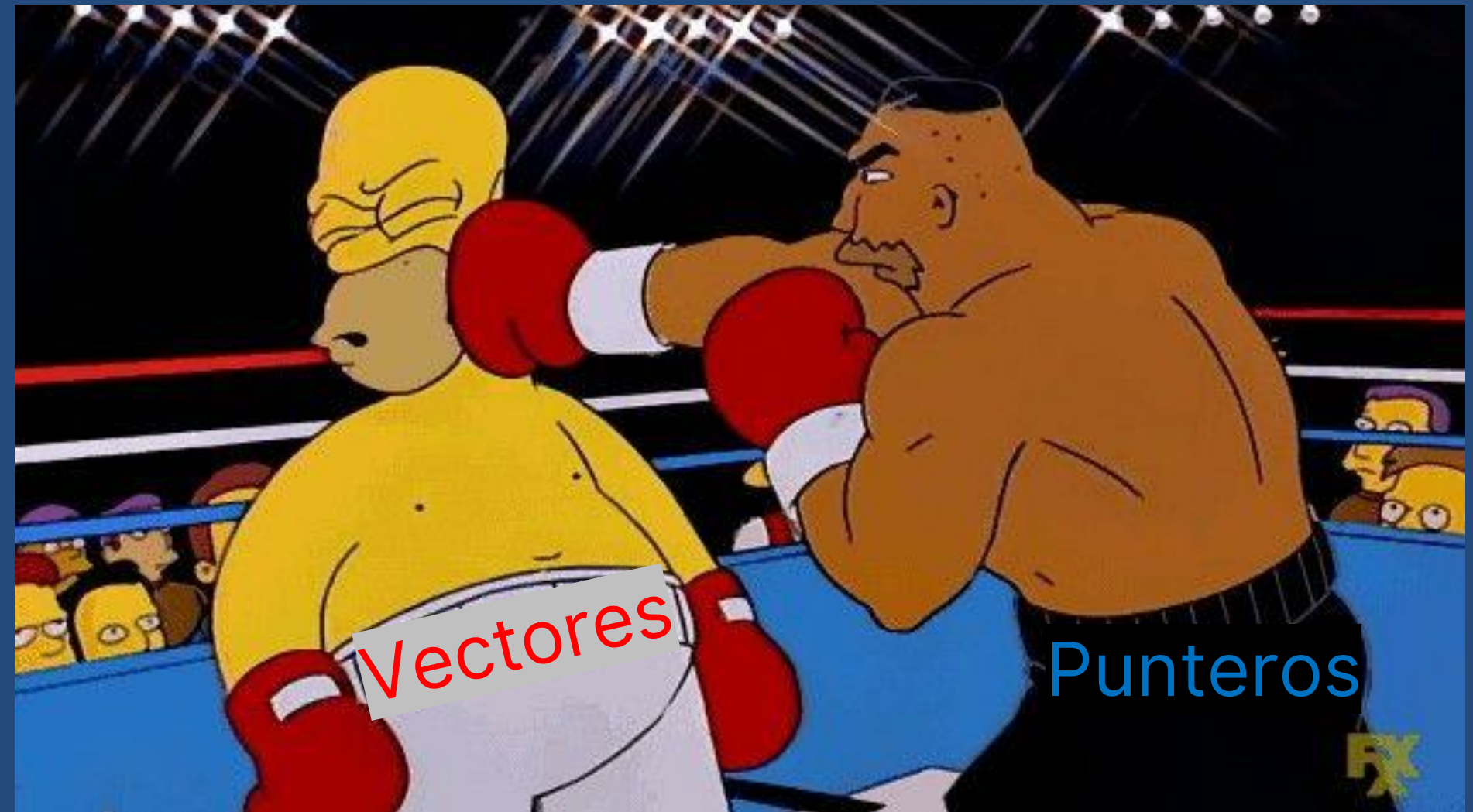
Conclusiones:

- La dirección de memoria que almacena el puntero no es siempre la misma.
- Al imprimir el puntero se imprime un número entero gigante.
- Se usa el sistema “Hexadecimal” (0 ... F).
- El ‘0x’ nos indica que el número está en base hexadecimal.

Punteros vs Vectores



destripando_vectores.c

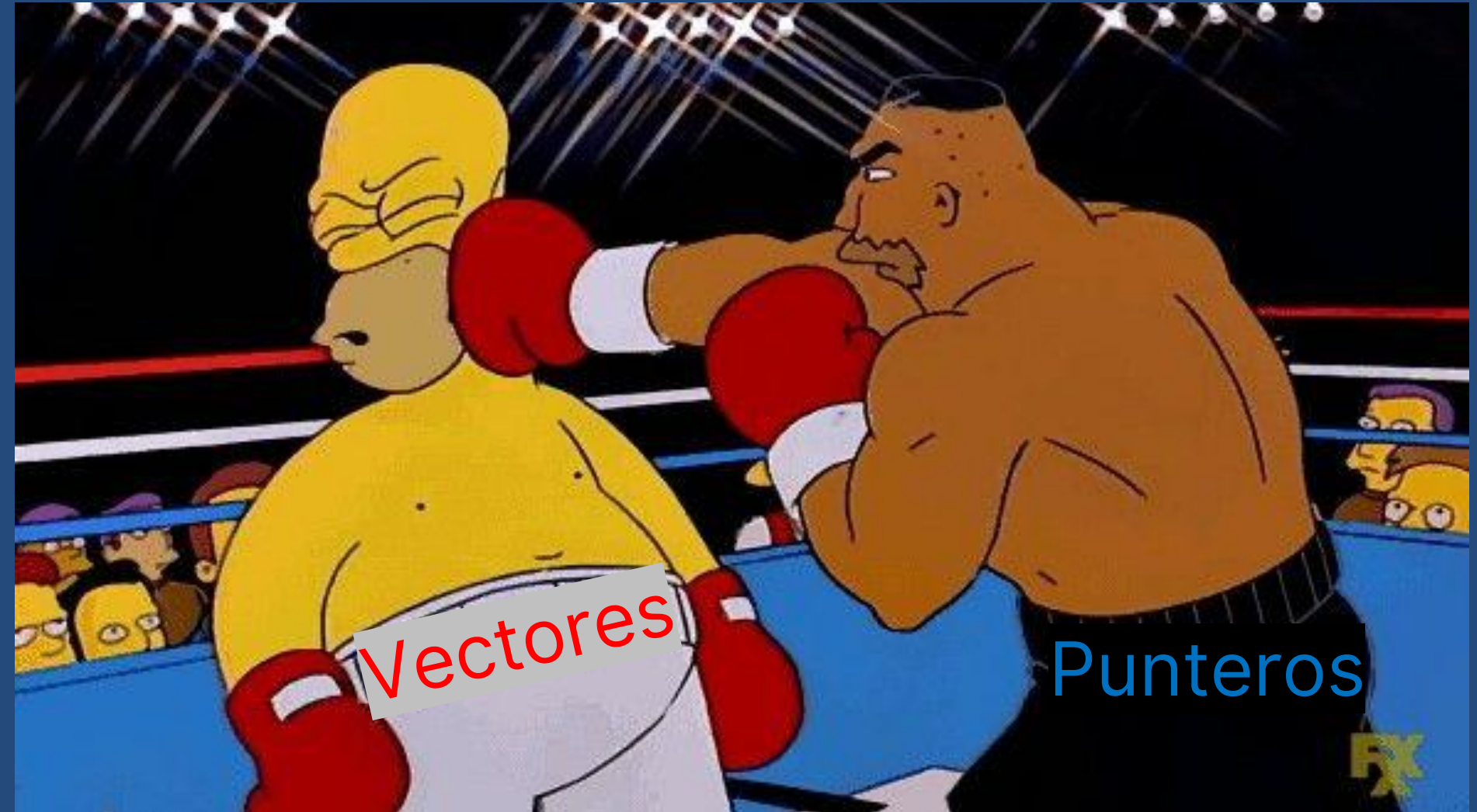


Punteros vs Vectores



destripando_vectores.c

- ✓ UN VECTOR ES UN PUNTERO
- ✓ EL NOMBRE DEL PUNTERO ACTÚA COMO PUNTERO CONSTANTE
- ✓ EL VECTOR APUNTA AL PRIMER ELEMENTO DEL ARREGLO
- ✓ LOS ELEMENTOS SE ENCUENTRAN CONTIGUOS EN MEMORIA
- ✓ ¿QUÉ DIFERENCIA A LOS STRINGS, DE OTROS VECTORES?

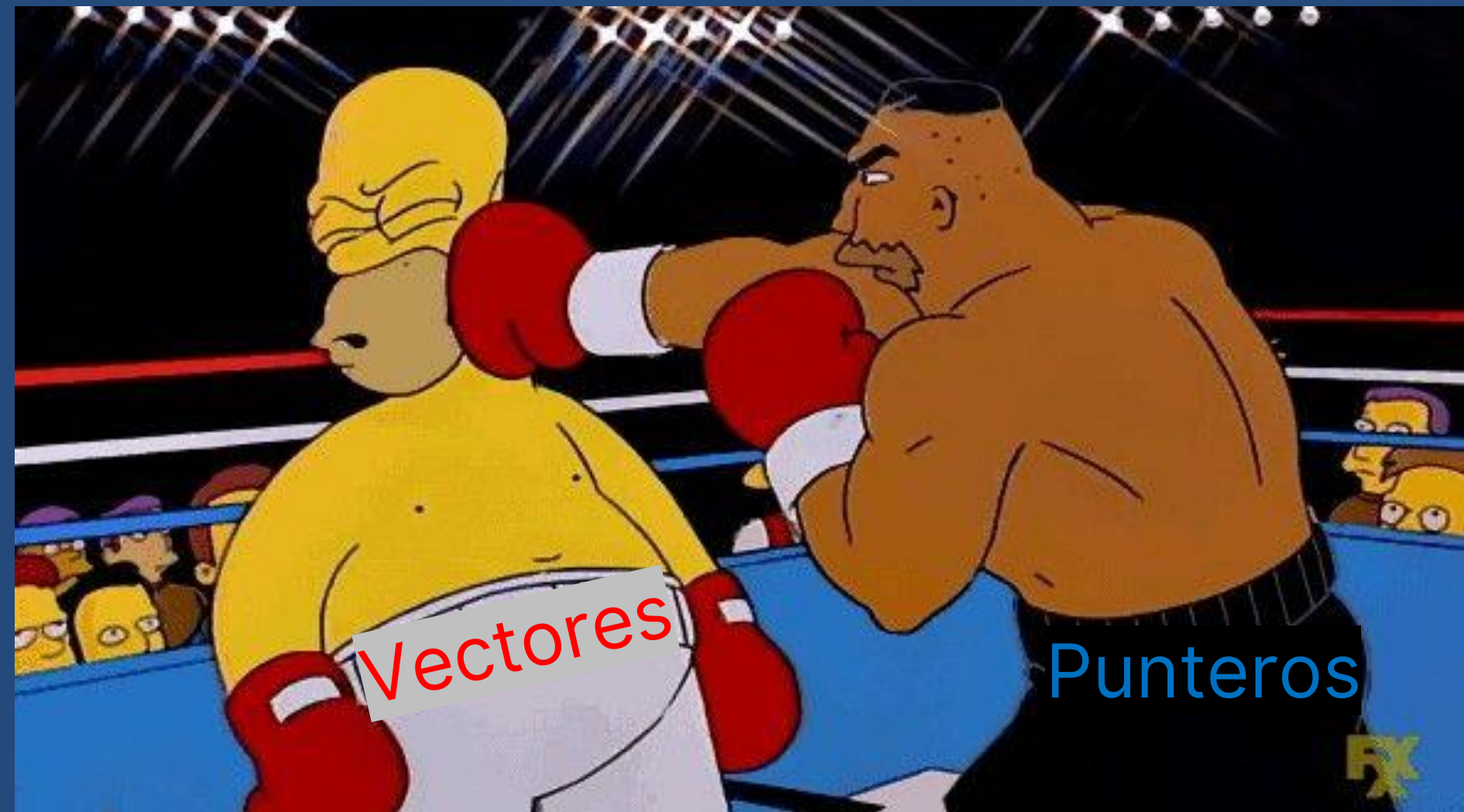


Punteros vs Vectores



destripando_vectores.c

- ✓ UN VECTOR ES UN PUNTERO
- ✓ EL NOMBRE DEL PUNTERO ACTÚA COMO PUNTERO CONSTANTE
- ✓ EL VECTOR APUNTA AL PRIMER ELEMENTO DEL ARREGLO
- ✓ LOS ELEMENTOS SE ENCUENTRAN CONTIGUOS EN MEMORIA
- ✓ ¿QUÉ DIFERENCIA A LOS STRINGS, DE OTROS VECTORES?



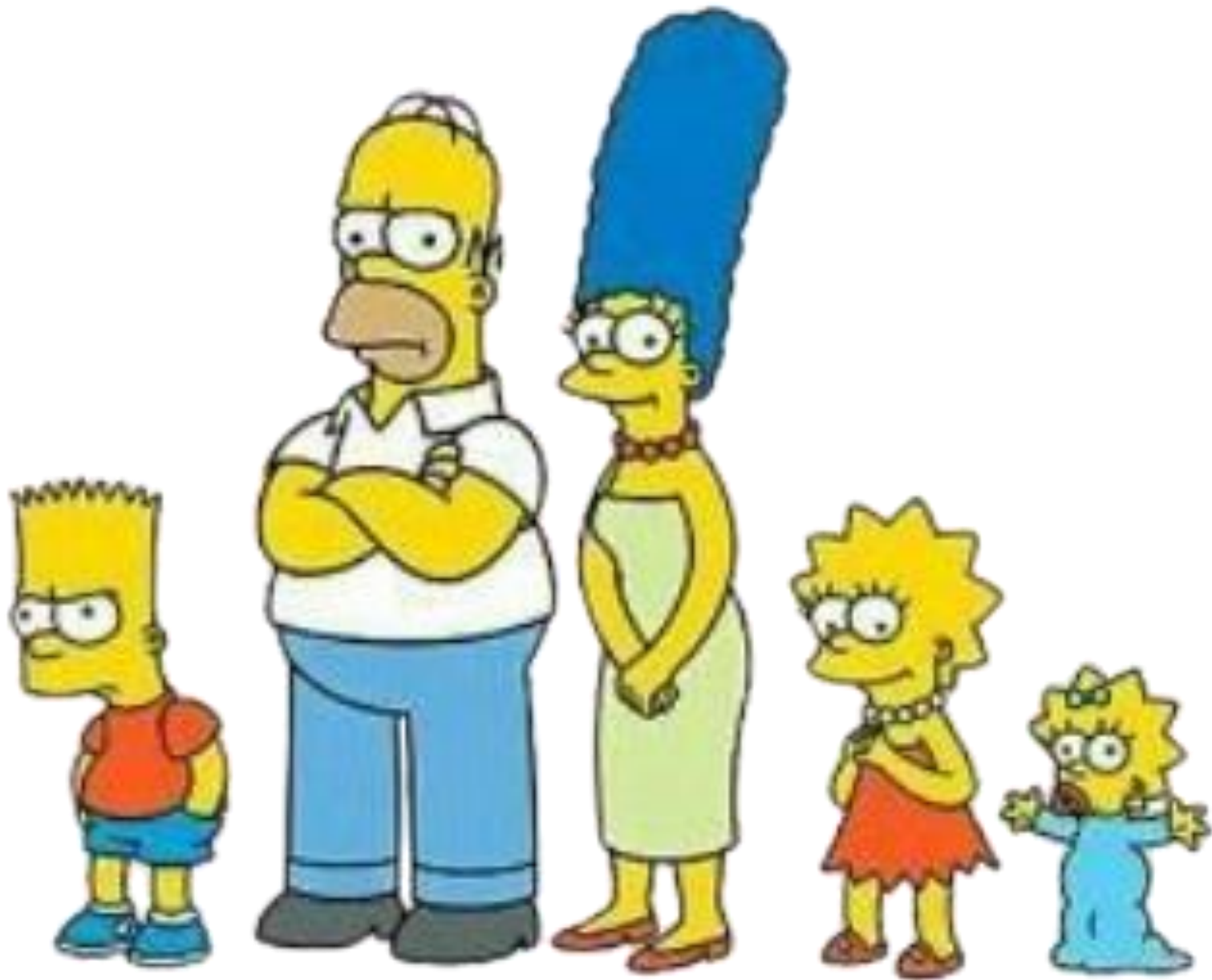
DE LOS STRINGS CONOCEMOS EL FINAL

HERRAMIENTAS DE APOYO

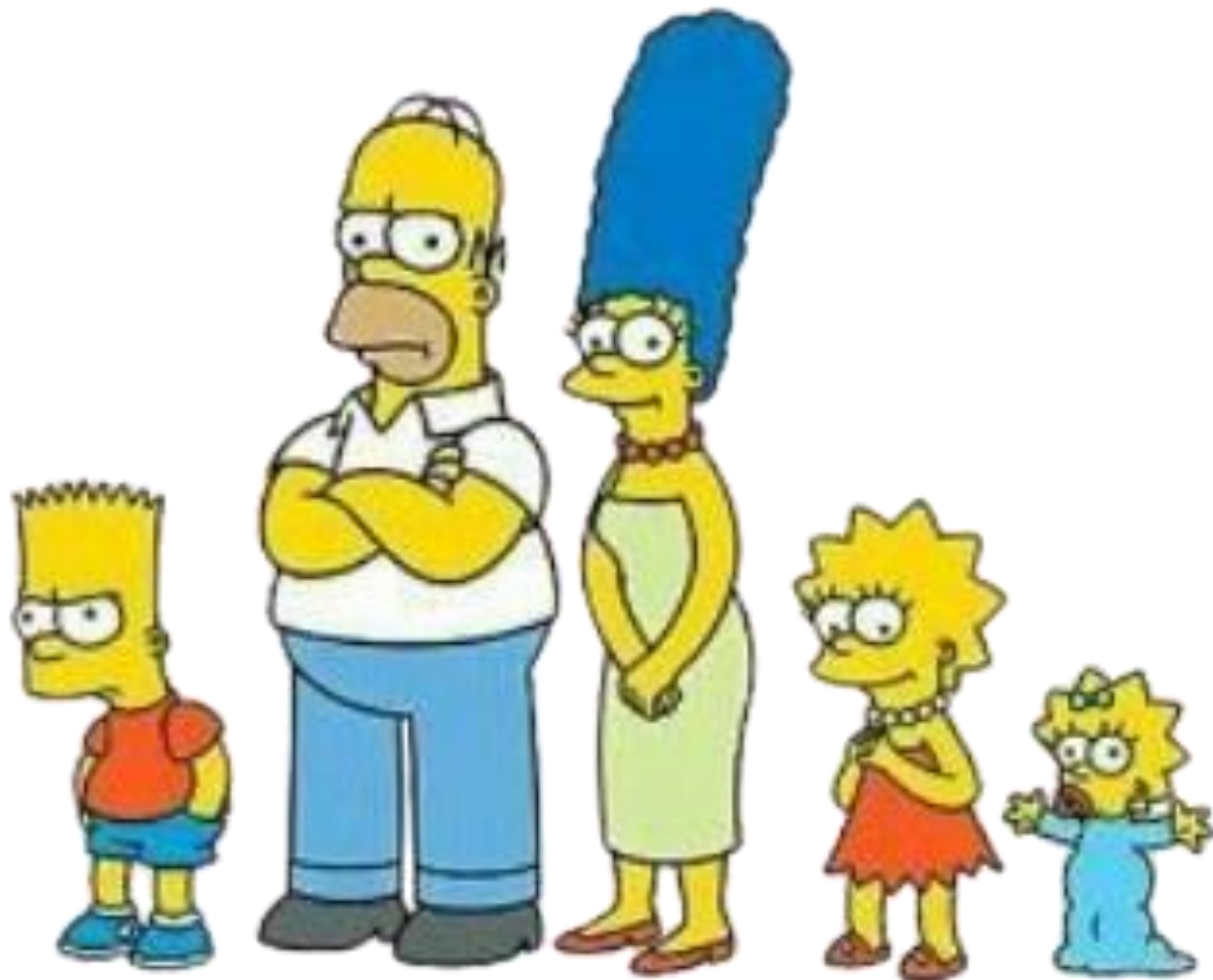
SIMULADOR DE MEMORIA RAM

CONVERSION DE HEXADECIMAL A DECIMAL

sizeof



SIZEOF



sizeof.c

- La función 'sizeof()' nos devuelve un 'size_t'.
- Para imprimir utilizamos '%zu'.
- Cada tipo tiene su cantidad respectiva cantidad de bytes.
- Los vectores ocupan (B x K):
 - B = Cantidad de Bytes del tipo de dato.
 - K = Cantidad de elementos.
- Todos los punteros ocupan lo mismo en memoria.
- En los structs existe el concepto de "Padding".

SEGMENTATION FAULT



- ¿Qué es?
 - Un mecanismo de defensa estricto del Sistema Operativo.
- ¿Qué lo causa?
 - Intentar acceder a una dirección de memoria “prohibida”.
- Caso típico:
 - Leer mas allá del límite de un vector.
- ¿En qué desemboca?
 - El Sistema Operativo aborta el proceso de inmediato.



segmentation_fault.c

```
Iniciando el programa...  
A punto de cometer un delito de memoria...  
Segmentation fault (core dumped)
```

FIN.

¿Alguna duda?

