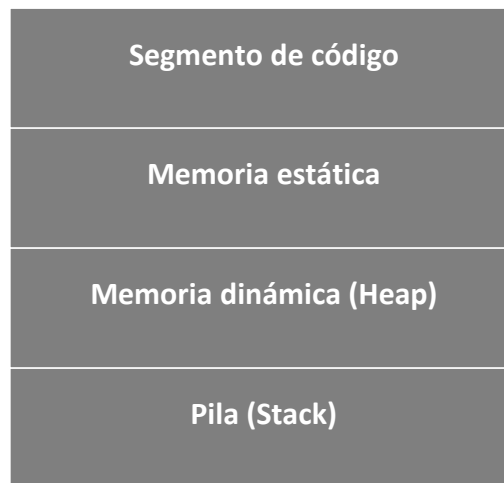


Punteros y asignación dinámica de memoria

Introducción

La asignación dinámica de memoria es una característica del lenguaje C. Consiste en asignar la cantidad de memoria necesaria para almacenar un tipo de dato durante la ejecución del programa, en vez de hacerlo en la fase de compilación de este. Cuando se crea un programa en el que es necesario manejar memoria dinámica el sistema operativo divide a la memoria en cuatro partes que son:



- **Segmento de código:** Almacena el código del programa.
- **Memoria estática:** Aquí se hace la reserva de los arreglos o matrices durante la compilación del programa.
- **Pila:** Se almacenan las llamadas a funciones recursivas.
- **Memoria dinámica:** Es la memoria que utiliza el programador para reservar espacio durante la ejecución del programa y luego la libera.

¿Cómo se accede a la memoria dinámica? Mediante los punteros.

Vamos a crear un puntero en memoria dinámica y asignarle un valor.

Para poder hacerlo, necesitamos de tres funciones en C:

La función **Malloc** permite reservar memoria. Se encuentra en la librería `stdlib.h` y está definida por:

`void *malloc (size_t size)`

Malloc devuelve un puntero genérico.


La función **Sizeof** permite calcular cuántos bytes se quieren reservar.

`sizeof (int) →` calcula cuantos bytes necesita un entero

La función **Free** permite liberar la memoria que se reservó con Malloc.

Entonces, vamos a crear un puntero `p` y asignarle el valor de 10.

```
int *p;  
  
p = (int *)malloc (sizeof(int));  
  
*p=10;
```



Reservamos memoria: malloc

¿Cuánta? `sizeof (int)`

Cómo la función malloc devuelve un puntero genérico (es decir, no apunta a nada) lo casteamos a que apunte a un entero (`int *`). Y esta dirección que devuelve malloc se la asignamos al puntero `p`.