Análisis y documentación de la memoria dinámica en C



Alumno: Luis Daniel Mendoza Rodríguez

Materia: Introducción a la programación

Mtro: Dra. María de los Ángeles Salazar Olmos

# Análisis y documentación de la memoria dinámica en C

## Memoria dinámica

La memoria dinámica es un mecanismo que permite al programador reservar y liberar memoria durante la ejecución del programa en vez de hacerlo estáticamente en tiempo de compilación.  
Por ejemplo cuando asignas un array con determinado longitud, al momento de compilación, se reserva cuanta memoria se requerirá para ese array, en base al tipo de dato que puede almacenar, pero la memoria dinámica permite que el espacio reservado para las variables cambie durante la ejecución y esto es útil si no sabes exactamente cuánto espacios necesitaras porque depende de información proporcionada por el usuario.  
  
La memoria dinámica se gestiona a través de la librería <stdlib.h> con funciones como malloc(), calloc(), realloc() y free()

## Introducción al video

El video muestra como se utiliza las funciones malloc() y free() para la gestión de la memoria dinámica, con un ejemplo sencillo de un array, cuya longitud será entregada por el usuario a través de scanf()

[](https://www.youtube.com/embed/-70pvCjWJDc?feature=oembed)

## Proceso del ejercicio:

1.- Inicio del boilarplate del programa

Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Aquí inicia el video diciendo que malloc sirve para solicitar asignar memoria, y recibe como parámetro la cantidad de memoria que quieres almacenar, en este caso se refiere al ejemplo, si quisieras almacenar un array con 4,000 espacios flotantes, no bastaría con pedirle 4,000 porque lo que recibe como parámetro es un valor en bytes, por lo que toma el tamaño de un float sizeof(float) y lo multiplica por la cantidad de espacios del array que quiere.

Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Luego explica que malloc regresa un puntero de void\*, que puede ser cualquier cosa, entonces lo que hay que hacer es guardar lo que malloc regresa en un puntero del valor que necesitamos y antes hacerle un cast (una conversión de tipo) por eso esl = (float\*) malloc….

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

En esta segunda parte, remplazo el 4,000 por una variable llamada longitud que permite al usuario indicarle al programa cuantos valores quiere que almacene, porque después de que el usuario ingrese el valor que se guarda en “longitud” luego se remplaza en la función malloc el 4,000, para asignar solamente los espacios en memoria que el usuario solicita

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Me falto un ;

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ejecución una vez arreglado el problema:  
Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

En esta ultima parte, agrega al final la liberación de memoria con la función free, aclara que una vez liberada la memoria no se puede acceder más delante de la historia.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Agregue comentario para organizar mejor. Y al final

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Comentarios finales

Respecto al ejercicio creo que es bastante claro y sencillo, lo que me llamo más la atención es que en lenguajes más modernos y de alto nivel como javascript, php y Python, esta gestión de la memoria corre por parte del interprete (que si es Python por ejemplo, es c) basta con declarar una variable de tipo array   
En PHP:  
$usuarios;  
$usuarios[] = [“nombre” => Luis Daniel, “apellido” => Mendoza]  
  
Al declarar $usuarios; se reservar un espacio en memoria que un no se conoce nada, y al hacer la asignación $usuarios[], le indicas que quieres que $usuarios sea un array y que guardaras algo en la siguiente posición disponible, en este caso 1.  
  
Sin embargo en C es necesario asignar el tamaño de la memoria en bytes y gestionarla porque la memoria dinámia se almacena en el heap, que no es especialmente grande.