GESTION DE MEMORIA ESTATICA

Estructura de datos

Conceptos fundamentales

Elkin Pabon - 2023

Definicion:

- La gestión de memoria estática se refiere al proceso de asignación y liberación de memoria en un programa de computadora durante la fase de compilación. La cantidad de memoria necesaria para variables, estructuras de datos y otros elementos es determinada y reservada antes de que el programa se ejecute. La asignación estática de memoria se realiza generalmente en tiempo de compilación y permanece constante durante la ejecución del programa.

CARACTERISTICAS

- **Asignación en Tiempo de Compilación:**

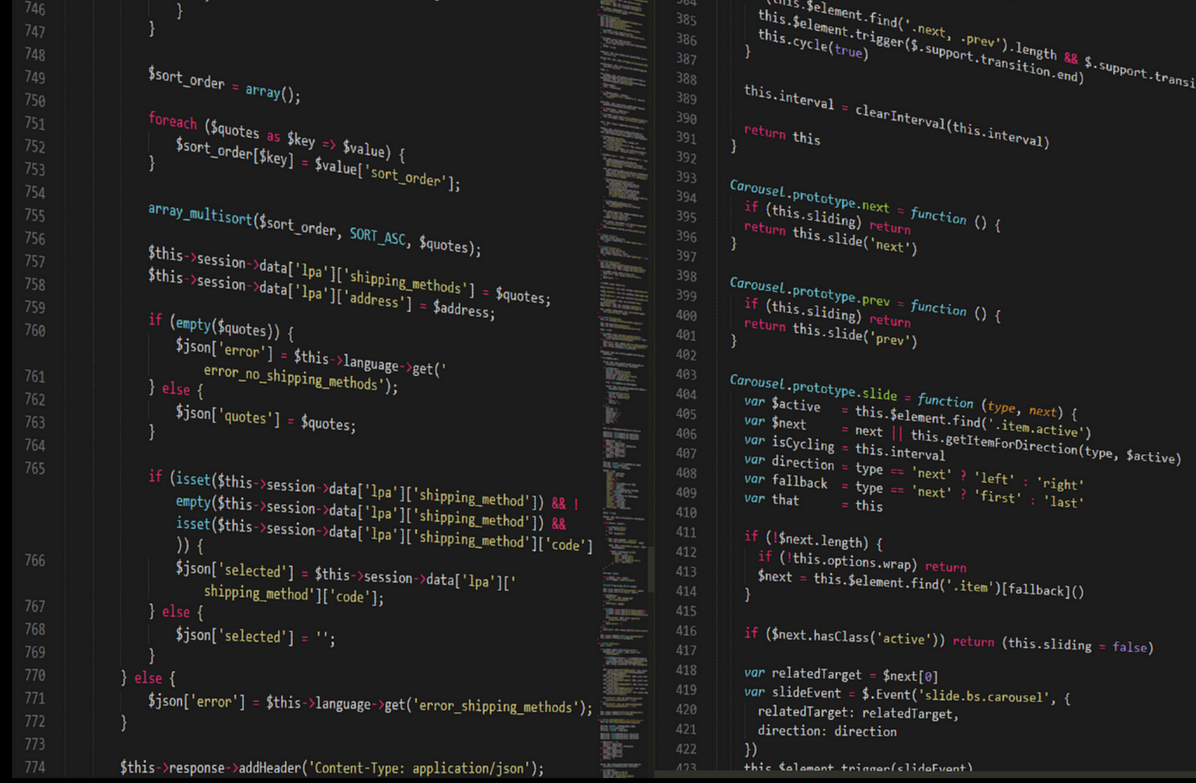
La memoria estática se asigna durante la fase de compilación del programa, antes de que se ejecute. Esto contrasta con la asignación dinámica, que ocurre en tiempo de ejecución.

- **Tamaño Fijo:**

El tamaño de la memoria asignada estátamente es fijo y se determina durante la compilación. No puede cambiar dinámicamente durante la ejecución del programa.

- **Eficiencia en Tiempo de Ejecución:**

La gestión de memoria estática tiende a ser más eficiente en tiempo de ejecución en comparación con la gestión dinámica. No hay gastos generales asociados con la asignación y liberación dinámica de memoria.



- **Acceso Rápido:**

Debido a que la asignación de memoria estática se realiza durante la compilación, el acceso a los datos almacenados estátamente es rápido y predecible. No hay necesidad de cálculos adicionales para encontrar la posición en memoria.

Ejemplo:

[*] main.cpp

```cpp
//Gestion de memoria estatica

#include <iostream>

using namespace std;

int main() {
    // Variable global
    int contadorGlobal = 0;

    // Función que utiliza memoria estática
    auto incrementarContadorLocal = [&]() {
        int contadorLocal = 0;
        ++contadorLocal;
    };

    cout << "Valor actual del contador global: " << contadorGlobal << endl;

    contadorGlobal += 5;

    // Llamando a la función que utiliza la variable local
    incrementarContadorLocal();

    // Utilizando nuevamente la variable global modificada
    cout << "Nuevo valor del contador global: " << contadorGlobal << endl;

    return 0;
}
```

(globals)

C:\Users\Andres\Desktop\main.exe

```
Valor actual del contador global: 0
Nuevo valor del contador global: 5

--------------------------------
Process exited after 0.07437 seconds with return value 0
Presione una tecla para continuar . . .
```