Maracaibo, 30/05/2022
Diego Rincón CI. 29929768
Estructuras Dinámicas de Datos

Asignación Dinámica de Memoria

Antes de definir la asignación dinámica de memoria tiene que estar claro que la asignación estática de memoria es fija y se reserva con un dato constante antes de la ejecución del programa, por lo que una vez asignada, no se puede cambiar. Por otro lado, la asignación dinámica de memoria puede ser solicitada durante la ejecución del programa y puede ser asignada, borrada o cambiada a voluntad del programador

Dando un ejemplo con arreglos, con asignación estática de memoria, los arreglos tienen un tamaño fijo y si llegan a almacenar menos datos que su tamaño, el resto de memoria que ocupan será desperdiciado, mientras que los arreglos asignados dinámicamente pueden tomar un valor asignado durante la ejecución del programa con el fin de almacenar los datos justos para su tamaño, logrando así no desperdiciar memoria

La asignación dinámica de datos es ideal cuando se busca eficiencia en un programa o cuando no se sabe de qué tamaño serán las variables.

Operadores

Para poder asignar memoria dinámicamente se necesitan dos operadores, los cuales trabajan haciendo uso de punteros. Estos son:

new: Reserva un número de bytes de memoria solicitados por una declaración, la cual debe tener el tipo de dato requerido.

```
int *p = new tipoDeDato[];
```

delete: Libera la memoria reservada por new, con el fin de obtener un uso eficiente y seguro.

```
delete[]p;
```

```
// Calculadora de notas, pidiendo el nro de notas al usuario

#include <stdio.h>

int main () {

    // Declaración de variables
    int cantidadNotas;
    int suma = 0;
    double resultado = 0;
    int *notas;

    // Pedir cantidad de notas al usuario
    printf ("Ingrese la cantidad de notas del estudiante: ");
    scanf ("%d", &cantidadNotas);

    // Creando un arreglo dinámico
    notas = new int [cantidadNotas];

    // Pidiendo las notas al usuario
    for (int i=0; i< cantidadNotas; i++) {
        printf ("Ingrese la nota [%d]: ", i+1);
        scanf ("%d", &notas[i]);
        suma += notas[i];
    }
```

```
// Mostrando notas y su posición en memoria
printf("Mostrando las notas del estudiante:\n");
for (int i=0; i<cantidadNotas; i++){
    printf("Nota [%d]: %d\n", i+1, notas[i]);
    printf("Posición en memoria: %d\n\n", &notas[i]);

resultado= suma/cantidadNotas;
printf("\n Promedio del estudiante: %lf", resultado);

// Liberando la memoria asignada
delete [] notas;

return 0;

}
```