

Universidad Rafael Urdaneta
Facultad de Ingeniería

Luis Romero 29.836.922

30 de Mayo de 2022

Estructura dinámica de datos

Investigar la asignación dinámica de arreglos
(uso de los operadores new y delete).

Un arreglo dinámico, es un array de elementos que crece o mengua dinámicamente conforme los elementos se agregan o eliminan. La forma de construir un array dinámico, es la asignación de espacio de tamaño fijo dividido en dos partes: La primera es la parte ocupada, y la segunda está libre para su crecimiento.

Así es sencillo añadir o eliminar elementos usando el espacio reservado.

Operador new

Denota una solicitud de asignación de memoria en Free Store. Con el tamaño del tipo de dato que vamos a ingresar, si hay suficiente memoria disponible, el nuevo operador inicia la memoria y devuelve la dirección de la recién asignada e inicializada a la variable de puntero.

Su sintaxis es la siguiente:

Variable - apuntador = new Tipo de dato

Para el caso de las matrices, es decir para un arreglo, el operador new también se usa para asignar un bloque de memoria de tipo data-type:

Variable - puntero = new tipo de datos [tamaño];

donde tamaño (una variable) especifica el número de elementos en una matriz

Operador delete

Cuando ya no necesitamos más el espacio reservado debemos liberarlo, es decir, indicar al ordenador que puede destinarlo para otros fines. Dado que es responsabilidad de desasignar la memoria asignada dinámicamente, se le proporciona el operador de delete. Al llamar al delete, el compilador primero llama al destructor y luego ejecuta delete.

Sintaxis:

delete - pointer - variable; o delete [pointer - variable];

Ejercicio/Ejemplo // El usuario determina un número, el programa devuelve números aleatorios

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int num, i;
    int * aleatorio;
```



```
printf("numero de elementos:");  
scanf("%d", &num);
```

```
aleatorio = new int[num];
```

```
for (i = 0; i < num; i++)  
{  
    aleatorio[i] = rand() % 100;  
}
```

```
printf("\n Numeros:");
```

```
for (i = 0; i < num; i++)  
    printf("%d", aleatorio[i]);
```

```
delete(aleatorio);
```

```
}
```