**Trabajo Práctico Nº 9.2**

**Marco Teórico:**





En este grafico tenemos dividido en sectores las partes de la memoria, tenemos 3 partes principales que vamos a explicar, datos globales, montón y pila. Este grafico se puede dividir en 3 subpartes. Los datos globales, el código del programa y el S.O(sistema operativo) forman parte de la memoria principal, el montón es parte de los datos dinámicos y la pila los datos locales.  
En los datos globales se van a almacenar todos los datos globales del programa, ósea toda la información por fuera del main y los subprogramas que se pueden utilizar por cualquier parte de nuestro código.  
En la pila se van a almacenar los datos locales de los subprogramas que son las variables o los parámetros que le pasamos y que nos devuelve.  
Y en el montón se almacenan todos los datos dinámicos, que son creados y destruidos después de su uso, la memoria tiene un sistema para controlar eso (SGMD) que es la encargada de liberar espacio en la memoria una vez que ya no se necesita.

1. Con el operador NEW nosotros reservamos memoria en el montón para un tipo específico y nos devuelve la primera dirección de memoria que se le asigna, esta tiene que ser asignada a un puntero para después poder usarla. El operador NEW nos permite también asignarle un valor a la variable que creamos, por ejemplo:  
   int \* p;  
   p = new int(12);.
2. El operador DELETE libera la memoria que nosotros reservamos para un puntero, si no lo hacemos nunca liberamos el espacio que estamos usando y podemos sobre cargar el sistema. De la misma forma que en el código un ( tiene que tener un ) para cerrarlo un NEW tiene que tener un DELETE para cerrarlo.  
   Si queremos liberar el espacio de memoria usado para el puntero anterior tenemos que hacer esto al final: delete p;.
3. Cuando nosotros creamos el puntero y lo definimos se le asigna un lugar en la PILA, después cuando creamos un NEW y se lo asignamos a ese puntero este se mueve al MONTON, no tenemos que olvidarnos de usar el DELETE para liberar ese espacio de memoria y liberar el MONTON.
4. Cuando vamos usando espacio en la PILA esta se llena para abajo y cuando usamos espacio en el MONTON esta va subiendo, si no liberamos el espacio después de usarlo se va a cargar hasta que no tenga mas espacio y colisionen. Es como tener una caja de libros y lo llenamos de izquierda a derecha y del otro lado de derecha a izquierda, si no vamos sacando libros en algún momento esa caja va a estar llena y no vamos a poder poner mas libros.
5. Su nombre lo dice, la pila tiene un espacio preasignado, si nosotros seguimos cargándole datos sin parar y sin liberarlos esta pila se va a desbordar y nos vamos a quedar sin espacio en la pila.
6. Existen varios errores comunes como el olvidarse de liberar un espacio de memoria, si no creamos un DELETE para liberar el espacio, el programa se va a cerrar sin errores pero nos va a quedar espacio de memoria usándose sin ninguna utilidad.  
   Tambien puede pasar que quieran usar un DELETE en un dato que no existe, crear la variable sin darle un NEW significa que no tiene espacio de memoria asignado entonces si tenemos mas DELETE que NEW vamos a recibir el error de que intentamos liberar un dato que no existe.  
   Podemos perder un dato dinamico cuando un dato de un puntero remplaza al dato de otro sin guardarlo, cuando remplazamos el dato 1 por el dato 2 el dato 1 se va a perder en el montón y como ya perdió la asignación al dato, vamos a querer borrar un dato que ya fue borrado.  
   El DELETE tiene que ir al final del todo, si usamos un DELETE p y después queremos mostrarlo en pantalla vamos a tener un error de que no se puede acceder a ese dato.
7. Los punteros ocupan muy poco espacio en la memoria, un array a datos dinámicos se refiere a que nosotros tenemos la estructura de nuestro array y usamos punteros para apuntar a datos dinámicos que van a llenar nuestro array. No tenemos que olvidar de usar el NEW y el DELETE en cada dato que creamos.
8. Un array dinamico es similar, pero la diferencia es que todo el array es creado y asignado a un puntero y después se utiliza un solo DELETE para liberar el espacio de todo el array, no tenemos que usar un NEW en cada dato y un DELETE en cada dato que utilizamos, solo uno al principio y al final.
9. Un array de datos dinámicos es un array lleno de punteros pero un array dinamico es un puntero apuntando a un array. Si bien el array de datos dinámicos es bueno en el sentido de que podemos crear un array sin limite de espacio ya que creamos y destruimos cada dato después de utilizarlo no es tan confiable a la hora de manejar datos si no lo controlamos bien, podemos borrar o remplazar datos y perderlos.  
   El array dinamico es mas ordenado y de fácil acceso además que es mucho mas simple el sistema de tener un solo NEW y DELETE al principio y al final, pero su desventaja esta en que cuando decidimos las dimensiones del array no lo podemos cambiar en tiempo de ejecución, decidimos la dimensión que va a tener el array, se reserva la memoria necesaria para eso y no se puede cambiar.