

Investigación y Reflexión, Listas Doblemente Encadenadas.

Las listas doblemente encadenadas es una estructura lineal de nodos en la cual cada uno de estos se encuentra conectado tanto hacia su siguiente posición como a la anterior, esto se logra a través de los apuntadores, lo cuales permiten que el contenido de una variable sea la dirección de otra variable, es por ello que como su nombre lo indica “apuntan” hacia una variable, de este modo se facilita el acceso a los diferentes valores ya que podemos hacer que un apuntador se mueva por las direcciones de memoria, y al obtener una de estas podemos modificar su contenido e incluso cambiar las direcciones a las que apunta, de tal manera que se nos permite manejar la memoria de una forma dinámica.

A partir de la teoría anterior ya se había analizado otra estructura, la de enlace simple hacia la siguiente posición, la cual también permite movernos entre nodos, sin embargo, podemos decir que lo lograba de una forma más rustica pues para generar un simple nodo en la última posición tenemos que movernos desde el primer nodo que tiene un apuntador inicial “head”, el cual permite llegar a cualquier lugar sin excepciones, de tal forma que para tener acceso a la posición final se tiene que recorrer toda la estructura; además en el hipotético caso de que se tuviera que devolver en 2 posiciones no se puede ya que no hay conexiones que logren esa forma de moverse, por lo que se tendría que iniciar nuevamente un apuntador temporal desde “head”. Es aquí donde entra la importancia y las ventajas de las listas doblemente encadenadas, pues al hacer que los nodos apunten tanto hacia adelante como atrás, podemos fijar apuntadores iniciales en cualquier lugar de la estructura sobre los que podemos movernos con libertad, de este modo creamos 2 apuntadores con dicha tarea, “head” (primer nodo de la estructura) y “tail” (último nodo), con los que podemos tener un acceso directo a los valores más alejados para crear un apuntador temporal y movernos según nos convenga. Regresando al ejemplo anteriormente mencionado, crear un nodo en la última posición, podemos ingresar directamente a “tail” y desde ahí generar otro nodo con las conexiones correctas y que además cuente ahora con el apuntador base “tail”; por otro lado, en caso de querer modificar la penúltima posición ya no sería necesario recorrer prácticamente toda la estructura de nuevo, sino que con el enlace de “tail” hacia posiciones anteriores se lograría tener acceso con el uso de un solo movimiento de un apuntador temporal iniciado en la última posición.

Este tipo de estructuras hace muy conveniente el acceso a cualquier nodo, sin embargo, es preciso mencionar que su complejidad al momento de generar un cambio de nodos en la estructura aumenta bastante, pues cambiar todas las conexiones y acomodarlas de la forma correcta se puede volver algo confuso. Es por ello que en la presente actividad se implementó el método de ordenamiento burbuja, el cual es bastante simple y permite un cambio de conexiones a través de la lógica, sin mucha dificultad y sin la necesidad de generar nuevos apuntadores temporales.

Dicho todo lo anterior, es más sencillo explicar el papel de las estructuras mencionadas en un problema de la naturaleza de la actividad. Si comparamos estrictamente las 2 estructuras, es indudable que las doblemente encadenadas tienen un mejor uso debido a la movilidad y acceso que existe entre nodos, si bien es más complejo el implementar los cambios de la estructura, no

es imposible, y es tarea del programador buscar la solución lógica para resolver este tipo de situaciones.

Cabe mencionar que, si bien este tipo de estructuras no es muy eficiente y funciona prácticamente igual que librerías como la de vector, pero con menos libertad, su entendimiento es muy importante ya que sienta bases hacia temas más complejos y genera un mayor pensamiento lógico para futuras estructuras, además genera una buena práctica de un tema muy importante y a menudo complejo para algunos estudiantes, los apuntadores.