

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
(Laboratorio 4)
Semestre 2014-1

Tema: Listas Enlazadas implementadas en lenguaje C

El manejo de memoria dinámica ha servido para implementar de forma eficiente muchas estructuras de datos. A su vez estas estructuras han servido para un manejo eficaz de la información.

La especificación de los tipos de datos facilita que su implementación no esté comprometida por los elementos empleados en su construcción. Es decir, la implementación sigue las definiciones de la especificación pero la forma cómo se han elaborado están ocultas al usuario de éstas, que normalmente es un programador.

Por tanto un objetivo de los tipos de datos es ocultar su implementación de modo que el usuario sólo debe recibir el perfil y la descripción del comportamiento de las operaciones que se pueden llevar a cabo con dicho tipo de dato.

En un lenguaje orientado a objetos es fácil lograr este objetivo, debido principalmente a que este paradigma propone el concepto de encapsular la data y sus métodos. En un lenguaje imperativo como el lenguaje C se puede lograr por lo menos una separación del código donde se implementan las estructuras, para que luego sean empleadas por cualquier usuario, sin necesidad de que el usuario sepa cómo están implementadas. El concepto al que estamos haciendo referencia es el de *librería*.

Existe la idea equivocada por parte de algunos alumnos, de considerar un archivo de cabecera (*.h) como una librería. El concepto de librería es otro y existen librerías que se enlazan en forma estática al momento de compilar el programa, así como librerías dinámicas que se comparte en tiempo de ejecución, este último trabajo es llevado a cabo por parte del sistema operativo.

Para este laboratorio necesariamente se llevará a cabo en Linux y el objetivo es que usted implemente las listas enlazadas y sus operaciones en una librería. En el laboratorio usted puede crear un programa (conteniendo *main*) para probar que las operaciones implementadas en su librería son correctas, sin embargo el JP evaluará su(s) librería(s) con otro programa cuyo objetivo será evaluar si la implementación es correcta.

A continuación se le proporciona algunos enlaces de Internet donde puede encontrar información acerca de como construir librerías en C.

<http://www.cs.dartmouth.edu/~campbell/cs50/buildlib.html>

http://www.cs.swarthmore.edu/~newhall/unixhelp/howto_C_libraries.html

http://www.adp-gmbh.ch/cpp/gcc/create_lib.html

http://wiki.osdev.org/Creating_a_C_Library

y cualquier otro enlace que encuentre en Internet le servirá no sólo para aprender el procedimiento, sino el concepto y su uso.

Hasta ahora en clase usted ha visto listas enlazadas como estructura más que como un tipo de dato. De hecho la implementación de un tipo de dato abstracto (TDA) normalmente es construida con listas enlazadas. Por ejemplo usted puede emplear listas enlazadas para implementar colas, pilas, listas (como TDA), árboles, etc.

El laboratorio estará formado en primer lugar por una parte (40%) del manejo de listas enlazadas con algunas operaciones, cuyo único objetivo es evaluar el conocimiento de dicha estructura. Y otra parte (60%) donde usted implementará tipos de datos abstractos empleando listas enlazadas. Solo en esta segunda parte se le pedirá que construya una librería.

A continuación una lista de ejercicios, en base a estos se le solicitará modificarlos o ampliar sus operaciones:

Parte 1 Dada una lista enlazada de enteros, implemente funciones y/o procedimientos para:

- a) Calcular el valor mayor de los datos e indicar la posición en que se encuentra (la lista no está ordenada y no se debe considerar ordenarla)
- b) Calcular el valor mínimo de los datos e indicar cuántas veces se encuentra en la lista.
- c) Que reciba una lista de enteros L y un número entero n de forma que modifique la lista mediante el borrado de todos los elementos de la lista que tengan este valor.
- d) Que reciba una lista de enteros y dos valores, el primero indica el valor que se debe reemplazar y el segundo indica el valor, por el cual se debe reemplazar.
- e) Que reciba una lista de enteros y dos enteros que indican alguna posición inicial y final de la listas, la función debe devolver una lista cuyos nodos deben comprender entre las posiciones antes indicadas.
- f) Que reciba dos listas ya ordenadas y devuelva una lista (diferente) también ordenada, sin aplicar ningún método de ordenación.

Parte 2 Implemente las siguientes TDA empleando listas enlazadas como una librería estática y como una librería dinámica en C.

Listas, con las siguientes operaciones

- Crear una lista vacía.
- Generar una lista unitaria formada por un elemento dado.
- Añadir un elemento por la izquierda.
- Añadir un elemento por la derecha.
- Consultar el elemento más a la izquierda.
- Consultar el elemento más a la derecha.
- Eliminar el elemento más a la izquierda.
- Eliminar el elemento más a la derecha.
- Determinar si una lista es vacía
- Concatenar dos listas.
- Calcular la longitud de una lista

Colas, con las siguientes operaciones:

- Crear una cola vacía
- Encolar un elemento
- Desencolar un elemento o servir el primer elemento de la cola.
- Consultar el primer elemento
- Determinar si una cola es vacía.

Pilas, con las siguientes operaciones:

- Crear la pila vacía
- Apilar un elemento.
- Desapilar un elemento.
- Consultar el elemento de la cima.
- Determinar si la pila está vacía

Para cada una de las operaciones emplee nombres apropiados.

En el laboratorio se le pedirá modificar cualquiera de estos tres TDA's, ampliando sus operaciones o definición. No se empleará otro tipo de dato abstracto que sea diferente a los mencionados.

Pando 23 de mayo del 2014.

Prof. Alejandro T. Bello Ruiz