

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

UNIVERSIDAD AUTONOMA METROPOLITANA

Programación Avanzada

Práctica 7:

“TDA – Lista Ligadas”

Fecha de Elaboración:

Martes, 5 de agosto del 2008

Fecha de Entrega:

Jueves, 14 de agosto del 2008

Contenido

1	Objetivos.....	2
2	Introducción.....	3
2.1	Ayuda complementaria.....	3
3	Desarrollo	4
3.1	Antecedentes: TDA Lista Ligada - Especificación	4
3.1.1	Conjunto de valores.....	4
3.1.2	Lista Ligadas – Implementación	4
3.1.3	Conjunto de operaciones.	5
3.2	Actividad 1 “Creación de un proyecto y utilización de la herramienta Make y definición de Lista mediante nodos”	5
3.2.1	Diseño Modular - Por Archivos.....	6
Actividad 2	“TDA – Lista Ligadas”	7
3.2.1	Definición del TDA Lista Ligada.....	7
3.2.2	Operaciones TDA Lista Ligada.....	7
3.2.3	Archivo make.	7
3.2.4	Menú TDA Lista Ligada	7
3.2.5	Diseño Modular	7

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

AVISOS:

- No se reciben reportes de las prácticas después de la fecha de entrega.
- En la página se encuentra el formato del reporte para las prácticas en donde se describen cada uno de los apartados que deberá considerarse. Se deberá seguir dicha plantilla para el desarrollo del reporte de las prácticas.
- Enviar a la cuenta de correo un paquete zipeado que contendrá:
 - Códigos fuentes (extensión **.c**).
 - Códigos Objetos (extensión **.o**)
 - Ejecutables (sin extensión)
 - En el reporte incluir por actividad (extensión **.doc**):
 - **Desarrollo** de la actividad.
 - **Resultados** de la actividad.
- En el subject del correo escribir: **Práctica 7 - <nombre completo>**.
- **Se entregará el reporte y los códigos fuentes impresos.**

1 Objetivos

- 1) Utilizar el diseño de programación ascendente y descendente en el enfoque de **programación modular** (enfoque “divide y vencerás”) para la resolución de problemas mediante el uso de **módulos** que serán traducidos a funciones (y/o procedimientos) en el lenguaje C.
- 2) Utilizar la herramienta **make** para optimizar y automatizar la creación de proyectos en C bajo el ambiente de Linux
- 3) Implementar la especificación del **TDA Lista Ligada** solicitando al sistema memoria para cada nodo de la lista.
- 4) Definir e implementar el **conjunto de valores** que especificaran el TDA Lista.
- 5) Implementar el **conjunto de operaciones** que permiten manipular el TDA.

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

2 Introducción

En el desarrollo de esta práctica se requiere que el alumno realice cada una de las actividades que se le indica.

2.1 Ayuda complementaria

- Referencia de la bibliografía contenida en la planeación del curso de Programación Avanzada.
- Notas del Curso de Introducción a al Programación 08-I.
- Guía Básica para el uso de C.
- La ayuda de la línea de comandos de Linux: *man make*.
- Practica 5 publicada en la siguiente página del curso programación avanzada.
URL:<http://uamisoft.izt.uam.mx/uamisoft/doku.php?id=amm:cursos:pa:practicasy:practica5>

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

3 Desarrollo

La sección del desarrollo de la práctica se divide en una serie de actividades que permiten englobar, desarrollar y alcanzar cada uno de los objetivos de la práctica. Es necesario leer con atención cada una de las actividades a desarrollar.

3.1 Antecedentes: TDA Lista Ligada - Especificación

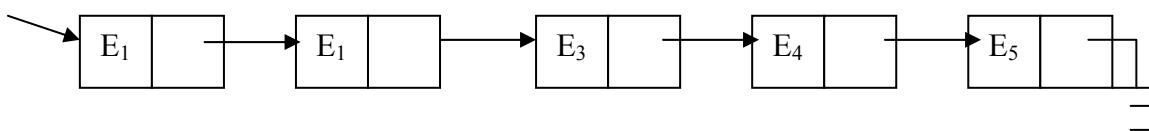
3.1.1 Conjunto de valores.

Una lista sobre un Dominio D es una secuencia a_1, a_2, \dots, a_n donde n es un número natural y $a_i \in D$.

Se dice que:

- El elemento a_i ocupa la posición i. Donde las posiciones i son tal que $1 \leq i \leq n$.
- a_i **precede** a a_{i+1} ($1 \leq i \leq n-1$)
- a_i **suced**e a a_{i-1} ($1 < i \leq n$)

3.1.2 Lista Ligadas – Implementación



Especificación del Nodo de la Lista:

```
typedef int TipoInfo;

typedef struct RegNodo nodo;

struct RegNodo
{
    TipoInfo elem;
    nodo *Siguiete;
};
```

Especificación del TDA Lista Ligada:

```
struct RegLista
{
    nodo *cabeza;
    int longitud;
};

typedef struct RegLista Lista;
```

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

3.1.3 Conjunto de operaciones.

- **Inserta (L, elemento, posicion)**
- **Elimina (L, posicion)**
- **Elimina (L, elemento)**
- **Busca (L, elemento)**
- **Longitud (L)**
- **Fin (L)**
- **Consulta (L , posicion)**
- **DesplegarListaIter (L)**
- **DesplegarListaRec (L -> cabeza)**
- **Es_Vacia (L)**
- **Es_Llena (L)**
- **Destruye (L)**
- **SumaNodos(L)**
- **SumaNodosRec(L)**
- **Haz_Nula (L)**
- **Primer (L)**
- **Resto (L)**

3.2 Actividad 1 “Creación de un proyecto y utilización de la herramienta Make y definición de Lista mediante nodos”

Con el desarrollo de esta actividad se alcanzará los objetivos 1 y 2 de la práctica.

Cree una carpeta referente a la practica y la actividad: /Practica7/Actividad1/ y cree los siguientes archivos:

ListaLigada.c

- **nodo *Crea_Nodo (elem)**
Recibe como parámetro un entero que representa el elemento que contendrá el nodo a crear.
- **Void DesplegarListaRec (cabeza)**
Recibe como parámetro un nodo y mediante la función recursiva envía a imprimir el contenido de una lista de nodos interconectados.

Error.c

- **void error (opcion)**
Envía a pantalla un mensaje de acuerdo al tipo de error, para el manejo de la lista_
- **void errorSys (opcion)**
Envía a pantalla un mensaje de acuerdo al tipo de error, para el manejo de instrucciones del sistema o memoria_

Principal.c

Desarrolla las pruebas para las funciones de Crear_Nodo y DesplegarLustaRec.

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

LibreriaPrueba,h

- *Especificación de librerías de uso común.*
- *Declaración de constantes*
- *Declaración y definición de tipo de datos nuevos*
- *Declaración de variables globales*
- *Definición de funciones por archivo de todo el proyecto*

Makefile

- Contienen las ‘instrucciones’ para automatizar la compilación y ligado de los archivos contenidos en un proyecto.

3.2.1 Diseño Modular - Por Archivos.

Desarrolle el diagrama modular.

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-P
Programación Avanzada - Practicas	Práctica 7

Actividad 2 “TDA – Lista Ligadas”

Con el desarrollo de esta actividad se alcanzará los objetivos del 1 al 5 de la práctica.

Cree un directorio, dentro del directorio Practica7, con el nombre ListaLigada. Reutilice los archivos generados en la actividad 1.

3.2.1 Definición del TDA Lista Ligada

Especifique la definición del TDA Lista Ligada en el archivo: LibTDAListaLigada.h

3.2.2 Operaciones TDA Lista Ligada

Desarrolle cada una de las operaciones con el manejo de listas.

3.2.3 Archivo make.

Desarrolle un archivo make para la compilación y ligado del proyecto.

3.2.4 Menú TDA Lista Ligada

Integre el uso de un menú que permita la interacción del usuario con las operaciones de manipulación del TDA Lista Ligada.

3.2.5 Diseño Modular

Desarrolle el diagrama modular al desarrollo de los módulos.