

Práctica 1: Programación en C

Especificaciones

- Esta práctica es individual.
- Debe de realizarse utilizando el lenguaje de programación C.
- Hacer un archivo `Makefile` para la compilación.
- Incluir un archivo `README.md` en donde se documente la práctica, con imágenes de ejemplos de ejecución.

Ejercicios

1. Lista enlazada

Crear el archivo `lista.c` e implementar una lista enlazada genérica, haciendo uso de estructuras y apuntadores (definir las estructuras y la firma de las funciones en el archivo `lista.h`):

```
struct nodo {
    void * elemento;
    struct nodo * siguiente;
};

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

Se deben de implementar las siguientes funciones:

```
void agrega_elemento(struct lista * lista, void * elemento);
```

Agrega un elemento al final de la lista.

```
void * obten_elemento(struct lista * lista, int n);
```

Permite obtener el n -elemento de la lista, empezando desde 0.

Regresa:

- El elemento número n de la lista.
- 0 si n está fuera del rango de la lista.

```
void * elimina_elemento(struct lista * lista, int n);
```

Elimina el n -elemento de la lista, empezando desde 0.

Regresa:

- El elemento eliminado.
- 0 si n está fuera del rango de la lista.

Recordar liberar la memoria asignada dinámicamente que ya no vaya a utilizarse.

```
void aplica_funcion(struct lista * lista, void (*f)(void *));
```

Aplica la función `f` a todos los elementos de la lista.

Ejemplo

Teniendo el programa:

```
#include "lista.h"
#include <stdlib.h>
#include <stdio.h>

void imprime(void * elemento) {
    char * e = (char*)elemento;
    printf("%c ", *e);
}

void mayuscula(void * elemento) {
    char * e = (char*)elemento;
    *e += 'A' - 'a';
}

int main() {
    struct lista lista = {cabeza:0, longitud:0};
    for(int c = 'a'; c <= 'k'; c++) {
        char * e = (char*)malloc(sizeof(e));
        *e = c;
        agrega_elemento(&lista, e);
    }
    printf("Longitud de lista: %d\nElementos: ", lista.longitud);
    aplica_funcion(&lista, imprime);
    aplica_funcion(&lista, mayuscula);
    printf("\nMasyúsculas: ");
    aplica_funcion(&lista, imprime);
    int n = lista.longitud;
    for(int i = 0; i < n; i++) {
        printf("\nElemento a ser eliminado: %c, ", *(char*)obten_elemento(&lista, 0));
        char * e = (char *)elimina_elemento(&lista, 0);
        printf("elemento eliminado: %c, longitud de lista: %d", *e, lista.longitud);
        free(e);
    }
    puts("");
}
```

se debe de tener la siguiente salida:

```
Longitud de lista: 11
Elementos: a b c d e f g h i j k
Masyúsculas: A B C D E F G H I J K
Elemento a ser eliminado: A, elemento eliminado: A, longitud de lista: 10
```

```
Elemento a ser eliminado: B, elemento eliminado: B, longitud de lista: 9
Elemento a ser eliminado: C, elemento eliminado: C, longitud de lista: 8
Elemento a ser eliminado: D, elemento eliminado: D, longitud de lista: 7
Elemento a ser eliminado: E, elemento eliminado: E, longitud de lista: 6
Elemento a ser eliminado: F, elemento eliminado: F, longitud de lista: 5
Elemento a ser eliminado: G, elemento eliminado: G, longitud de lista: 4
Elemento a ser eliminado: H, elemento eliminado: H, longitud de lista: 3
Elemento a ser eliminado: I, elemento eliminado: I, longitud de lista: 2
Elemento a ser eliminado: J, elemento eliminado: J, longitud de lista: 1
Elemento a ser eliminado: K, elemento eliminado: K, longitud de lista: 0
```

2. Manejo de archivos

Crear el archivo `archivos.c` e implementar un programa que despliegue el siguiente menú en un ciclo:

```
1. Insertar archivo
2. Leer archivo
3. Eliminar archivo
4. Imprimir archivos
```

Se debe de leer la opción que ingrese el usuario desde la entrada estándar y realizar las siguientes funciones:

- La primera opción debe de leer el nombre de un archivo desde la entrada estándar y guardarlo en una lista enlazada (utilizando la implementación del ejercicio anterior).
- La segunda opción debe de leer un número n desde la entrada estándar y desplegar el contenido del archivo número n en la lista, si se trata de un directorio, desplegar los nombres de los archivos del directorio.
- La tercera opción lee un número n de la entrada estándar y elimina el elemento n de la lista.
- La cuarta opción imprime los elementos de la lista, es decir, los nombres de los archivos.

Imprimir mensajes de error en la salida de error estándar (si el archivo no existe, los índices están fuera de rango, se ingresaron opciones inválidas, etc.)

Ejercicios extra

e1. Bubble Sort con apuntadores

Válido sólo para quienes asistieron al laboratorio

Crear un programa que lea una lista de números enteros desde la línea de comandos, le asigne dinámicamente un espacio de memoria a cada entero y guarde los apuntadores a éstos en un arreglo. Utilizar el algoritmo de ordenación Bubble Sort para ordenar los números por medio de sus apuntadores en el arreglo.

Ejemplo

```
[usr@srv]$ ./bubblesort 42 2 13 19 0 142 41 53 31
Arreglo:    42 2 13 19 0 142 41 53 31
Ordenado:   0 2 13 19 31 41 42 53 142
```

e2. `/etc/passwd`

Crear un programa que lea un archivo que tenga el formato de `/etc/passwd` y muestre el significado de los

campos de cada usuario. Debe de leer el nombre del archivo desde la línea de comandos.

Ejemplo

Si `/etc/passwd` tiene el siguiente contenido:

```
root:x:0:0:root:/root:/bin/bash
ricardo::1000:1000:Usuario ricardo:/home/ricardo:/bin/sh
ramon:$1$kmIB05Wr$s2Ty91SbBvFJQLaGqc/LG/:1001:1001:Usuario ramon:/home/ramon:/bin/dash
ruben:$5$qbgHgwwba6q7yGUHF$9G6iaR/qGUoENb1AczFoFvzvEYA6rMIFxn8ljtjiA5:1002:1002:Usuario ruben:/home/ruben:/bin/fish
```

Entonces el programa debe de tener la siguiente salida:

```
[usr@srv]$ ./lee_passwd /etc/passwd
Nombre de usuario: root
Contraseña: Contraseña en archivo /etc/shadow
UID: 0
GID: 0
GECOS: root
Home: /root
Shell: /bin/bash

Nombre de usuario: ricardo
Contraseña: Autenticación sin contraseña
UID: 1000
GID: 1000
GECOS: Usuario ricardo
Home: /home/ricardo
Shell: /bin/sh

Nombre de usuario: ramon
Contraseña: $1$kmIB05Wr$s2Ty91SbBvFJQLaGqc/LG/
UID: 1001
GID: 1001
GECOS: Usuario ramon
Home: /home/ramon
Shell: /bin/dash

Nombre de usuario: ruben
Contraseña: $5$qbgHgwwba6q7yGUHF$9G6iaR/qGUoENb1AczFoFvzvEYA6rMIFxn8ljtjiA5
UID: 1002
GID: 1002
GECOS: Usuario ruben
Home: /home/ruben
Shell: /bin/fish
```

Se sugiere utilizar `strtok` o `strsep`.