

PRACTICA NO. 4

Manejo dinámico de estructuras

Fecha de elaboración: 13-Sept-2018

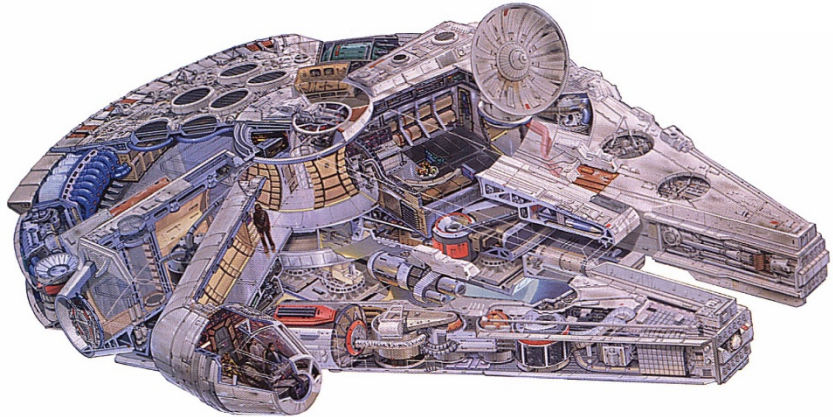
Objetivo

Estudiar los conceptos de creación dinámica de estructuras y dinámica de arreglos a estructuras

Introducción: Estructuras y manejo dinámico de memoria

Las estructuras son un tipo de datos compuesto, ya que las variables de éste tipo son capaces de almacenar un conjunto de valores. A diferencia de los arreglos las estructuras permiten almacenar elementos de diferente tipo.

Por ejemplo, si deseamos almacenar los datos de un alumno: número de lista, edad y promedio entonces podríamos definir una estructura que contuviera tres elementos: dos enteros y un flotante, cada uno de estos elementos les denominamos campos de la estructura:



TAlumno

numLista (int)	edad (int)	promedio (float)	<i>campo</i>
10	23	9.8	<i>contenido</i>

Para definir una estructura emplearemos la sentencia **typedef**, y normalmente la colocamos al principio del programa (fuera de cualquier función) o la definimos en algún archivo de encabezados (.h)

```
typedef struct {  
    int numLista;  
    int edad;  
    float promedio;  
} TAlumno;
```

La sentencia anterior define que una variable del tipo **TAlumno** poseerá los campos **numLista**, **edad** y **promedio** cada uno del tipo especificado.

Una variable del tipo **TAlumno** se declara y se emplea como se muestra:

```

int main() {
    TAlumno alumno;
    alumno.numLista = 10;
    scanf("%d",&alumno.edad);
    printf("Promedio: %d\n",alumno.promedio);
}

```

Un arreglo de alumnos, sigue las reglas de cualquier arreglo, a continuación se ejemplifica un grupo de alumnos modelado como un arreglo. A continuación se muestra el código de un programa para leer y desplegar los datos de un grupo de alumnos:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct {
5      int numLista;
6      int edad;
7      float promedio;
8  } TAlumno;
9
10 int main()
11 {
12     int max,n;
13     TAlumno *g;
14
15     int k;
16
17     printf("Numero maximo de alumnos? ");
18     scanf("%d",&max);
19     g = (TAlumno *)malloc(sizeof(TAlumno)*max);
20     printf("Numero de alumnos actualmente inscritos? ");
21     scanf("%d",&n);
22
23     for (k=0;k<n;k++) {
24         g[k].numLista = k+1;
25         printf("\n\nAlumno: [%2d]\n",g[k].numLista);
26         printf("edad? ");
27         scanf("%d",&((g+k)).edad);
28         printf("Promedio? ");
29         scanf("%f",&g[k].promedio);
30     }
31
32     printf("\n\nEstado actual del grupo: \n");
33     for (k=0;k<n;k++) {
34         printf("[%d] edad: %2d prom: [%5.2f]\n",g[k].numLista, g[k].edad, ((g+k)).promedio);
35     }
36
37     return 0;
38 }
39

```

Actividades

Nivel básico

Una librería requiere un sistema de cómputo que le permita administrar los libros que tiene a la venta. La información que guarda es la siguiente:

- Película (30 caracteres)
- Director:
 - Nombre (30 caracteres)
 - Apellido (30 caracteres)
- Actor Principal
 - Nombre (30 caracteres)
 - Apellido (30 caracteres)
- Número de ejemplares (entero)
- Precio (flotante)
- **Monto invertido (flotante)**



El monto invertido se calcula como: el producto de precio por el número de ejemplares.

Ejercicio 1. Crear un programa en el

- 1) Declare una estructura llamada PELICULA fuera de cualquier función

que en el main() :

- 2) cree dos variables del tipo **apuntador** a Película
- 3) Cree de manera dinámica mediante **malloc()** dos estructuras, una para cada apuntador
- 4) Lea los datos de ambas películas excepto el monto invertido
- 5) Calcule el monto invertido como el producto de precio por número de ejemplares
- 6) Imprima los datos de ambas películas indicando el monto invertido en cada una de ellas, y al final indicar cuál es el total invertido en ambas películas.

Ejercicio 2. Hacer un programa

- 1) Declare una estructura llamada PELICULA fuera de cualquier función

que en el main() :

- 2) Preguntar al usuario cuantas películas desea ingresar,
- 3) Se debe crear de manera dinámica un arreglo para almacenar todas las películas, por medio de un apuntador llamado **catalogo** utilizando la función **malloc()**
- 4) **A partir de este punto manejar el apuntador como si fuera un arreglo**
- 5) Leer los datos de cada película y almacenarlos en el arreglo creado anteriormente
- 6) Una vez leídas todas las películas, calcular para cada película el monto invertido.
- 7) Una vez calculados los montos totales invertidos, calcular el monto total invertido como: la suma los montos invertidos en cada película, y guarde el resultado en una variable llamada sumaTotal

- 8) Imprimir el catálogo completo de las películas, el cual debe incluir además de la información el monto invertido en esa y al final indicar el monto total invertido.

Ejercicio 3. Hacer un programa

- 1) Declare una estructura llamada PELICULA fuera de cualquier función
que en el main() :
- 2) Preguntar al usuario cuantas películas desea ingresar,
- 3) Se debe crear de manera dinámica un arreglo para almacenar todas las películas, por medio de un apuntador llamado **catalogo** utilizando la función **malloc()**
- 4) **A partir de este punto manejar el apuntador como tal, es decir no se permite que en el programa exista alguna expresión con []**
- 5) Leer los datos de cada película y almacenarlos en el arreglo creado anteriormente
- 6) Una vez leídas todas las películas, calcular para cada película el monto invertido.
- 7) Una vez calculados los montos totales invertidos, calcular el monto total invertido como: la suma los montos invertidos en cada película, y guarde el resultado en una variable llamada sumaTotal
- 8) Imprimir el catálogo completo de las películas, el cual debe incluir además de la información el monto invertido en esa y al final indicar el monto total invertido.

Ejercicio 1 De casa

Ejercicio 1. Crear un programa en el que:

- 1) Declare una estructura llamada PELICULA
- 2) Escriba una función llamada **lectura()** que lea los datos de una película, y la información que lee la coloca en la estructura cuya dirección recibe como parámetro
- 3) Escriba una función **calcula()** que recibe como parámetro la dirección de una película y calcula el monto invertido en esa película.
- 4) Escriba una función **escritura()** que reciba una estructura del tipo Pelicula y que imprima todos los datos de esa estructura
- 5) En la función **main()**:
 - a. Cree dos variables del tipo **apuntador** a Pelicula
 - b. Cree de manera dinámica mediante **malloc()** dos estructuras, una para cada apuntador
 - c. Utilice la **lectura()** para leer los datos de cada estructura
 - d. Utilice la función **calcula()** para determinar el monto invertido en cada película
 - e. Determine el monto total invertido en ambas películas
 - f. Utilice la función **imprime()** para presentar la información de cada una de las dos películas e imprimir al final el monto invertido



Ejercicio 2. Hacer un programa que:

- 1) Declare una estructura llamada PELICULA
- 2) Con base a la función **lectura()** del ejercicio 1 de casa cree una función llamada **leerCatalogo()** que reciba como parámetro un arreglo de películas y el número de películas de este catálogo y solicite al usuario todas las películas del catálogo.

- 3) Con base en la función **calcula()** del ejercicio 1 de casa cree una función llamada **calculaInversion()** que reciba como parámetro un arreglo de películas y el número de películas de este catálogo y calcule para cada película el monto invertido. La función debe regresar como resultado de su llamada el monto total invertido en todo el catálogo.
- 4) Con base en la función **escritura()** del ejercicio 1 de casa cree una función llamada **imprimeCatalogo()** que reciba como parámetro un arreglo de películas y el número de películas de este catálogo e imprima el catálogo completo en pantalla
- 5) En la función **main()**:
 - a. Cree una variable tipo del tipo **apuntador** a Pelicula
 - b. Pregunte al usuario de cuantas películas consta su catalogo
 - c. Cree de manera dinámica mediante **malloc()** un arreglo para almacenar las películas
 - d. Utilice la **leerCatalogo()** para leer los datos del catálogo
 - e. Utilice la función **calculaInversion()** para determinar el monto invertido en cada película y almacenar el monto total invertido en el catalogo
 - f. Determine el monto total invertido en ambas películas
 - g. Utilice la función **imprimeCatalogo()** para imprimir todas las películas del catálogo y al final imprimirá el monto invertido

Ejercicio 3 de casa (opcional, nivel aprendiz de hacker –nivel intermedio-)

Como puede observar cada vez que se requiere pasar como parámetro el catálogo también se necesita pasar el número de películas que contiene el catálogo, además el monto invertido es un valor particular de cada catálogo.

Una alternativa para manejar estos elementos como una sola unidad es crear una estructura CATALOGO que contenga tres elementos:

un apuntador al arreglo de películas,

una variable que contenga el número de películas

y una variable que contenga el monto total invertido.

Declare una estructura de este tipo y haga las modificaciones necesarias para que del ejercicio 2 de casa funcione pasando a las funciones leerCatalogo(), calculaInversion() e imprimeCatalogo() un solo parámetro del tipo struct CATALOGO *

Reto (Para volverse loco -nivel Avanzado-):



Suponga que se tiene una red de tiendas de películas y que cada tienda tiene su propio catálogo, por lo que se quiere tener un arreglo de catálogos de películas, como las compañía crece constantemente, se desea que el arreglo de catálogos de películas sea dinámico, es decir al principio se le pregunta al usuario cuantos catálogos desea, y se crea el arreglo justo para ese número de catálogos. Con base en el Ejercicio 3 de casa resuelva el problema. **¡Que la fuerza te acompañe!**

