
Llenguatges de Programació

Curs 2006 – 07 / Bloc #2 - Seminari #3

Activitat

NO ESCRIVIU RES EN AQUEST FULL!!

En aquesta activitat fareu una nova versió del programa que gestiona una base de dades de pel·lícules que va fer al seminari passat. Recordeu que el programa permetia afegir pel·lícules, veure totes les dades que hi havia a la base de dades i sortir.

En aquesta nova versió enlloc de guardar les dades en una taula dinàmica ho fareu en una llista.

Seguiu els passos que es van donant. Heu d'escriure en fulls apart els codis dels EXERCICIS.

/* EXERCICI 1:

Definiu un tipus de dada que es digui *node_pellicula* i que tingui els camps *Nom*, *Any* i *seguent* (que ha de ser un apuntador a una estructura de tipus *node_pellicula*) **/*

/ Declaració de funcions i procediments. Els que tenen un nombre al darrera els haureu de codificar **/**

```
void crear_llista_buida (); /*3*/
int carregar_dades (); /*5*/
void llegir_pellicula (node_pellicula *);
void menu ();
int es_buida ();
void mostra_pellicules();
int afegir_ultim (node_pellicula *); /*4*/
int eliminar_primer (); /*6*/
void desar_dades (); /*7*/
```

/* EXERCICI 2:

Declareu una variable (que serà global, com a cas excepcional) que es digui *primer* i que sigui un apuntador de tipus *node_pellicula*. Aquest apuntador sempre contindrà l'adreça del primer element de la vostra llista **/*

/* Programa principal */

```
void main()
{
    node_película aux;
    int control,opcio;

    crear_llista_buida();
    control = carregar_dades();

    if(control == 0)
        printf("Les dades no s'han carregat correctament. \n");
    else
    {
        do
        {
            opcio = menu();
            switch (opcio)
            {
                case 1: omplir_película(&aux);
                    afegir_ultim(&aux);
                    break;
                case 2: control = es_buida();
                    if (control == 0)
                        printf("No hi ha informació.");
                    else
                    {
                        mostra_películes();
                    }
                    break;
                case 3: desar_dades();
                    break;
                default:printf{"Opció incorrecta.\n"};
                    break;
            }
        }while (opcio != 3);
    }
}
```

/* codi dels procediments i les funcions : a continuació fareu el codi d'alguns dels procediments i funcions que utilitza aquest programa. Us adjuntem els corresponents algorismes que heu vist a classe de AP i el que heu de fer és traduir-los a C */

EXERCICI 3:

procediment que crea una llista buida

/* Crear una llista buida */

procediment CrearLlista ()

primer ← NUL

fiprocediment

EXERCICI 4:

Funció per afegir un element al final de la llista

```
/* Afegir un element al final de la llista */
funció AfegirÚltim (node_película punter aux ): lògic
var
    NouElement, ultim: punter node_película
    Correcte: lògic
fivar
si Assignar(NouElement) = NUL aleshores
    Correcte ← FALS /* Retorna fals si no es pot crear el nou element */
sino
    *NouElement.info ← aux.info /* omplim camp a camp la informació */
    *NouElement.següent ← NUL
    /* Si la llista està buida el nou element serà el primer */
    si primer = NUL aleshores
        primer ← NouElement
    sino
        /* Buscar l'últim element */
        ultim ← primer
        mentre *ultim.següent ≠ NUL
            ultim ← *ultim.següent
        fimentre
        *ultim.següent ← NouElement /* Enllacem el nou element amb l'últim */
    fisi
    Correcte ← CERT /* Retorna cert si s'ha pogut afegir el nou element */
    fisi
retorn Correcte
fifunció
```

EXERCICI 5:

Funció per carregar les dades a una llista dinàmica. Seguiu els següents passos:

- obrir el fitxer dades.txt per llegir, on suposarem que hi ha la informació de la base de dades en el següent format:

ET 1982 The Lord of the rings: The fellowship of the ring 2001 ...
--

MENTRE no arribem al final del fitxer...

- o llegir una pel·lícula i guardar-ne les dades en una variable de tipus node_película
 - o afegir la pel·lícula al final de la llista, utilitzant el procediment afegir_ultim
- tancar el fitxer

EXERCICI 6:

Procediment per eliminar el primer element de la llista

```
/* Eliminar el primer element de la llista */
funció EliminarPrimer ( ): lògic
var
    aux: punter node_película
    Correcte: lògic
fivar
si primer = NUL_aleshores
    Correcte ← FALS /* Retorna fals si no hi ha cap element a la llista */
sino
    aux ← primer
    primer ← *primer.següent /* Actualitza el primer element */
    Alliberar (aux) /* Allibera l'element eliminat */
    Correcte ← CERT /* Retorna cert si s'ha pogut eliminar el primer element */
fisi
retorn Correcte
fifunció
```

EXERCICI 7:

Procediment per desar les dades de la llista en el fitxer dades.txt. S'ha de fer el procediment de l'exercici 5 de forma inversa. Seguiu els següents passos:

- obrir el fitxer dades.txt per escriure

MENTRE no arribem al final de la llista

- o escriure en el fitxer la informació del primer element de la llista
- o eliminar el primer element de la llista (utilitzeu la funció eliminar_primer)