

Sistemes Operatius II - Pràctica 3

Octubre 2018

La tercera pràctica se centra en en la persistència de l'estructura de l'arbre, és a dir, en poder desar i carregar la informació emmagatzemada a/de disc de forma binària.

Índex

1	La pràctica	2
1.1	Interfície de menú	2
1.2	Creació de l'arbre	2
1.3	Emmagatzemament de l'arbre a disc	2
1.4	Consulta d'informació emmagatzemada a l'arbre	3
2	Implementació	3
3	Entrega	4

1 La pràctica

La pràctica 2 s'ha centrat en la lectura de fitxers en format text. Aquesta pràctica se centrarà en la manipulació de fitxers binaris, formalment coneguts com a fitxers no formatats.

En concret, la pràctica 3 se centra en.

- Utilitzar una interfície de menú senzilla que permeti que l'usuari interactui amb l'aplicació, veure secció 1.1. En aquesta pràctica es proporciona a l'estudiant del codi C d'aquesta interfície.
- La persistència de l'arbre. Per tal d'evitar haver de crear l'arbre cada cop que s'inicialitza l'aplicació, es proposa poder emmagatzemar i carregar l'estructura d'arbre a/de disc quan l'usuari vulgui, veure secció 1.3.

Es descriuen a continuació cadascun dels anteriors punts.

1.1 Interfície de menú

Per tal de facilitar la interacció de l'usuari amb l'aplicació es proposa desenvolupar una interfície de menú textual. El menú ha d'incloure les següents 5 opcions:

1. Creació de l'arbre. La creació de l'arbre correspon a la pràctica 2. En cas que hi hagi un arbre prèviament creat caldrà alliberar la memòria associada. Veure secció 1.2
2. Emmagatzemament de l'arbre. En seleccionar l'usuari aquesta opció l'aplicació haurà de demanar per teclat el nom del fitxer on es desarà l'arbre. A continuació l'aplicació desarà en aquest fitxer la informació dels nodes l'arbre mitjançant funcions d'entrada/sortida no formatada. A la secció 1.3 s'especifica el format de fitxer en què s'emmagatzema l'arbre.
3. Lectura de l'arbre. En seleccionar l'usuari aquesta opció l'aplicació haurà de demanar per teclat el nom del fitxer amb l'arbre a llegir. L'aplicació llegirà del fitxer fent i crearà l'estructura d'arbre, veure secció 1.3. En cas que hi hagi un arbre prèviament creat caldrà alliberar la memòria associada.
4. Permetre fer consultes respecte la informació emmagatzemada a l'arbre, veure secció 1.4. Aquesta funcionalitat ja s'ha implementat a la pràctica 2.
5. Sortida. Mitjançant aquesta opció s'allibera tota la memòria reservada i se surt de l'aplicació.

1.2 Creació de l'arbre

En seleccionar l'usuari aquesta opció l'aplicació haurà de demanar per teclat el fitxer amb la llista d'aeroports així com el fitxer de dades. L'aplicació crearà l'arbre i en finalitzar tornarà a mostrar el menú amb totes les opcions.

1.3 Emmagatzemament de l'arbre a disc

L'emmagatzemament de l'arbre a disc evita haver de crear l'arbre cada cop que s'engegui l'aplicació. Mitjançant el menú es podrà decidir si es vol desar o carregar la informació de l'arbre a disc. La informació s'haurà de desar mitjançant la tècnica no formatada, fent servir el format de fitxer

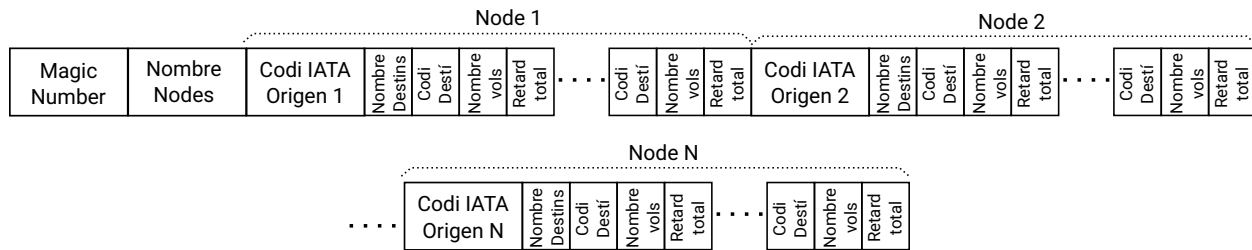


Figura 1: Format del fitxer per emmagatzemar la informació emmagatzemada a l'arbre.

especificat a la figura 1. S'especifica el format de fitxer per assegurar interoperabilitat entre els formats generats per totes les aplicacions dels vostres companys.

Aquest és el detall de l'estructura del fitxer. Tots els sencers que s'especifiquen són amb signe.

- Nombre màgic, 4 bytes. El nombre màgic del fitxer és 0x0133C8F9.
- Nombre de nodes N de l'arbre, 4 bytes, un sencer.
- Per a cada node s'emmagatzema la següent informació: a) el codi IATA de l'aeroport origen (sense el byte 0 de la cadena), b) nombre d'elements que hi ha a la llista enllaçada dels destins del node, c) la llista enllaçada formada pels seus elements: el codi IATA destí (sense el byte 0 de la cadena), el nombre de vols associats a aquell destí i el retard total (o la informació que vosaltres hagueu emmagatzemat a cada element de la llista).

No cal invertir els bytes per assegurar compatibilitat entre arquitectures big i little-endian. Els nodes es poden emmagatzemar a disc en l'ordre que es vulgui (i.e. es pot començar pel node arrel o bé pels nodes fills). Observar que al fitxer només es desa la informació emmagatzemada a cada node. No s'hi desa informació associada a l'estructura de l'arbre (pares, fills o germans de cada node).

En llegir la informació de l'arbre es llegirà el fitxer en el format especificat anteriorment. Atès que només es llegeix la informació de cada node és possible que l'estructura de l'arbre resultant sigui diferent de l'estructura que hi havia en el moment de desar-lo. Això no importa pas ja que l'estructura de l'arbre no té importància per a aquestes pràctiques.

En llegir l'arbre, comprova que el nombre màgic és correcte. Si no ho és, imprimeix un missatge d'error i sortiu de l'aplicació. En cas contrari, podeu llegir el fitxer suposant que totes les dades estan en el format correcte.

1.4 Consulta d'informació emmagatzemada a l'arbre

L'usuari haurà de poder realitzar les dues consultes implementades a la pràctica 2, és a dir,

1. Donat un aeroport origen (que s'introdueix per teclat), imprimir per pantalla els retards dels destins associats.
2. Cercar l'aeroport que té més aeroports destins.

2 Implementació

Aquesta secció dona alguns consells per tal d'implementar correctament aquesta pràctica.

Es proporciona a l'estudiant del codi en C del menú a utilitzar. No s'ha inclòs el codi associat a cadascuna de les opcions possibles. Tingueu en compte que l'usuari pot escollir les opcions del menú en qualsevol ordre i, per tant, caldrà tenir en compte les restriccions associades. Per exemple, a l'hora de desar l'arbre cal assegurar que hi ha un arbre a memòria; a l'hora de llegir un arbre de disc cal alliberar l'arbre en cas que n'hi hagi algun carregat a memòria, etc.

Es proposa seguir el següent ordre a l'hora d'implementar la pràctica

1. Implementeu la funció que permet crear l'arbre fent servir els dos paràmetres que fan falta (els aeroports d'origen i el fitxer de dades). El codi ha de funcionar per al fitxer complet de la base de dades (de 7 milions de línies).
2. La consulta de la informació associada a l'arbre.
3. Un cop arribat aquí, assegureu-vos que tot funciona bé amb el **valgrind**.
4. Implementació de les funcions per desar i carregar la informació de l'arbre a disc. L'arbre es desarà en un únic fitxer tal com s'ha comentat a la secció 1.3. Per a desar l'arbre es recomana fer servir la instrucció *fwrite* i per llegir la instrucció *fread*, veure fitxa del campus.
5. Un cop hagueu implementat l'escriptura i lectura de l'arbre de disc, assegureu-vos que funciona correctament fent servir el **valgrind**. Comproveu també que en llegir un fitxer de disc s'imprimeixen per pantalla les mateixes estadístiques que s'imprimeixen en crear l'arbre directament amb l'opció 1.

Es proporciona amb aquesta pràctica la implementació realitzada pel professorat. Podeu fer servir aquest codi per fer proves. El fitxer **tree.so2** ha estat creat amb el fitxer de dades que se us ha entregat i amb el format especificat a la figura 1.

3 Entrega

El fitxer que entregueu s'ha d'anomenar **P3_NomCognom1NomCognom2.tar.gz** (o **.zip**, o **.rar**, etc), on **NomCognom1** és el nom i cognom del primer component de la parella i **NomCognom2** és el nom i cognom del segon component de la parella de pràctiques. El fitxer pot estar comprimit amb qualsevol dels formats usuals (**tar.gz**, **zip**, **rar**, etc). Dintre d'aquest fitxer hi haurà d'haver tres carpetes: **src**, que contindrà el codi font, i **doc**, que contindrà la documentació addicional en PDF. Aquí hi ha els detalls per cada directori:

- La carpeta **src** contindrà el codi font. S'hi han d'incloure tots els fitxers necessaris per compilar i generar l'executable. El codi ha de compilar sota Linux amb la instrucció **make**. Editeu el fitxer *Makefile* en cas que necessiteu afegir fitxers C que s'hagin de compilar. El codi font ha d'estar comentat. El codi ha de funcionar per al fitxer complet de la base de dades (de 7 milions de línies).
- El directori **doc** ha de contenir un document (tres o quatre pàgines, màxim cinc pàgines, en format PDF, sense incloure la portada) explicant el funcionament de l'aplicació, la discussió de les proves realitzades i els problemes obtinguts. En aquest document no s'han d'explicar en detall les funcions o variables utilitzades.

La data límit d'entrega està indicat al document de planificació. El codi té un pes d'un **80%** (codi amb funcions comentades, codi modular i net, ús correcte del llenguatge, bon estil de programació, el programa funciona correctament, tota la memòria és alliberada, sense accessos invàlids a memòria, etc.). Tingueu en compte que els professors comprovaran el bon funcionament del vostre codi fent servir el **valgrind**. El document té un pes del **20%** restant.