

Sistemes Operatius II - Pràctica 2

Setembre del 2018

Índex

1	Introducció	2
2	Les eines a utilitzar	2
2.1	Llibreria C d'arbre binari balancejat	2
2.2	Llibreria C d'una llista enllaçada	2
3	La pràctica	3
3.1	Lectura de la informació dels aeroports d'origen	3
3.2	Lectura del fitxer de vols	3
3.3	Computació d'estadístiques	4
3.4	Alliberament de la memòria	4
4	Implementació	4
5	Entrega	4

1 Introducció

En aquesta pràctica es comença a desenvolupar el codi del projecte. Com s'ha comentat a la pràctica anterior, l'objectiu general del projecte pràctic és desenvolupar una aplicació (sense interfície gràfica) que permeti extreure i indexar informació dels vols dels Estats Units. Els fitxers de text a processar són proporcionats al campus tot i que també es poden baixar lliurement d'Internet, veure <http://stat-computing.org/dataexpo/2009/the-data.html>. En aquest web es poden trobar una sèrie de fitxers amb informació dels vols de diferents anys. Cada fitxer és un fitxer de text pla i conté la informació dels vols d'un determinat any. Cada fila de cadascun d'aquests fitxers conté la informació d'un vol. Entre altres coses, conté informació sobre l'aeroport d'origen, l'aeroport destí, el retard en sortir, el retard en arribar, etc.

En aquest projecte només farem servir el fitxer de dades de l'any 2008. El fitxer complet conté més de 7 milions de línies, tot i que de moment, per facilitar la feina, es farà servir un fitxer amb 10.000 línies.

2 Les eines a utilitzar

2.1 Llibreria C d'arbre binari balancejat

Observeu la figura 1. És un exemple d'arbre binari que conté cadenes de caràcters que corresponen a codis IATA (International Airport Abbreviation Code) d'aeroports. Per a cada node, en aquest arbre, s'emmagatzemen a l'esquerra les paraules que lexicogràficament són anteriors a la paraula del node, mentre que a la dreta es troben les paraules lexicogràficament posteriors a la paraula del node.

En aquesta pràctica se us proporciona d'un codi en C d'un arbre binari balancejat que està preparat per indexar valors sencers, no pas cadenes. Proveu d'executar el codi exemple. Caldrà modificar el codi per tal d'adaptar-lo a les necessitats de la pràctica, és a dir, que l'arbre permeti indexar cadenes de caràcters (en particular, els codis IATA dels aeroports).

2.2 Llibreria C d'una llista enllaçada

Es proporciona també una implementació d'una llista enllaçada genèrica indexada per un determinada variable. A l'exemple que es proporciona en aquesta pràctica la llista està indexada per sencers i permet comptar el nombre de vegades que cada sencer apareix. Caldrà modificar també aquest codi per adaptar-lo a les necessitats de la pràctica.

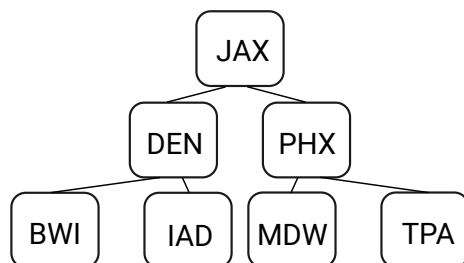


Figura 1: Arbre binari indexats per l'aeroport d'origen.

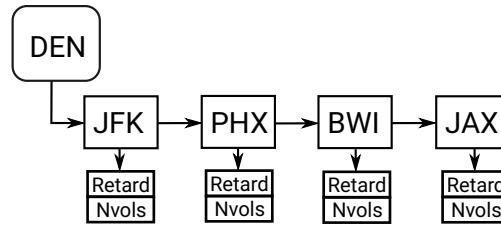


Figura 2: Llista enllaçada de cada node de l'arbre. El node de l'arbre correspon a l'aeroport d'origen, mentre que la llista enllaçada està indexada per l'aeroport destí i emmagatzema el retard total així com el nombre de vols associats.

3 La pràctica

La pràctica se centra en desenvolupar una aplicació C que té tres arguments: el fitxer de dades de vols, la llista d'aeroports i un aeroport origen.

```
$ practica2 <dades.csv> <aeroports.csv> <origen>
```

L'aplicació ha de realitzar les següents operacions

3.1 Lectura de la informació dels aeroports d'origen

Es proporciona, juntament amb aquesta pràctica, d'un fitxer amb els codis IATA dels aeroports d'origen. Cada codi IATA té tres caràcters. El format d'aquest fitxer és similar al que s'ha fet servir a la primera pràctica: a la primera línia s'indiquen el nombre d'elements del fitxer i, a continuació, es llisten els elements. Els codis dels aeroports d'origen es faran servir per tal de construir l'arbre. A l'hora de llegir el fitxer assegureu-vos d'eliminar el '\n' del final de la cadena atès que es demana (veure més endavant) que les dades es llegeixin amb la funció *fgets*.

3.2 Lectura del fitxer de vols

El fitxer de vols, que també anomenarem fitxer de dades, conté les dades a extreure amb els vols d'avió. Observeu que el format del fitxer: la primera línia és la capçalera, mentre que la resta de línies contenen les dades separades per comes (per això el format del fitxer s'acostuma a anomenar Comma Separated Values). Suposar que no se sap d'avant-mà quantes línies de dades conté el fitxer.

Per aquest projecte es demana extreure de cada fila la següent informació

- Columna 15: retard d'arribada a l'aeroport destí, en minuts. Aquest valor és un nombre sencer i pot ser positiu o negatiu. Un "NA" és indicació de "Not Available".
- Columna 17: aeroport d'origen, en codi IATA.
- Columna 18: aeroport destí, en IATA.

Cada node de l'arbre, que correspon a l'aeroport d'origen, haurà d'emmagatzemar en una llista enllaçada els aeroports destins juntament amb el retard total i el nombre de vols computats a partir del fitxer de dades, veure figura 2.

Una forma habitual d'extreure subcadenaes d'un fitxer de dades és la funció *strtok*. En aquest projecte es recomana no fer servir aquesta funció ja que a) No té el funcionament esperat en cas que

hi hagin dades en què hi ha dues comes juntes, per exemple, “,”. La funcionalitat de la funció *strtok* no permet extreure la cadena buida entre les dues comes. b) La funció *strtok* no està preparada per funcionar amb múltiples fils, funcionalitat que es farà servir a les pràctiques 4 i 5. Per aquestes raons es recomana implementar una funció pròpia que permeti extreure subcadena entre dues comes. Podeu suposar que la subcadena té una longitud limitada (per exemple, inferior a 100 caràcters).

3.3 Computació d'estadístiques

Un s'han llegit totes les dades, es demana

1. Computar, per a l'aeroport d'origen especificat com a segon argument a l'hora d'executar l'aplicació, el retard mig per cadascun dels seus destins. Imprimir els resultats per pantalla.
2. Computar quin és l'aeroport que té més destinacions. Imprimir el resultat per pantalla.

3.4 Alliberament de la memòria

Un cop s'han imprès les estadístiques per pantalla, allibereu tota la memòria dinàmica que s'ha reservat. Assegureu-vos a més que tot funciona correctament fent servir el *valgrind*. A la primera fitxa del campus teniu informació sobre el funcionament d'aquesta aplicació.

4 Implementació

Es proposa a continuació una forma de procedir per fer la pràctica

1. Executeu el codi d'exemple de l'arbre binari balancejat així com el codi de la llista enllaçada.
2. Implementeu el codi proposat a la secció 3. Aneu per passos, no intenteu implementar tot el codi de cop. Per exemple, comenceu per implementar només la secció 3.1, llegint les dades amb la funció *fgets* i assegurant que l'aplicació *valgrind* no dona errors d'execució ni informa de problemes d'alliberament de memòria. A continuació implementeu la secció 3.2, llegint les dades amb la funció *fgets* i assegurant de nou un bon funcionament del codi amb el *valgrind*. Finalment, implementeu la funcionalitat d'estadístiques de la secció 3.3.

Disposeu en aquesta pràctica del codi executable implementat pels professors de l'assignatura. Així podeu comprovar si el vostre codi funciona correctament.

5 Entrega

El fitxer que entregueu s'ha d'anomenar `P2_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 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 dues carpetes: `src`, que contindrà el codi font, i `doc`, que contindrà la documentació en PDF. Aquí hi ha els detalls per cada directori:

- La carpeta **src** contindrà el codi font comentat (només fa falta que ho estiguin les funcions). 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**. Assegureu-vos que el codi no dóna problemes en ser executat amb el **valgrind**. Utilitzeu la funció *fgets* per llegir les dades dels fitxers.
- El directori **doc** ha de contenir un document (tres o quatre pàgines màxim, en format PDF, sense incloure la portada) en què s'inclouï:
 - Comenteu els camps membres de l'estructura del node de l'arbre. Indiqueu quines funcions heu modificat (i com les heu modificat) per tal de permetre que l'arbre binari indexi per cadenes de caràcters. Feu el mateix pels camps membres de la llista indexada.
 - Indiqueu algun dels resultats que obteniu amb el fitxer que se us proporciona amb la pràctica. Proveu, per exemple, amb els aeroports de DEN (el corresponent a Denver) o ATL (Atlanta).

El codi té un pes d'un **70%** (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.). El document i les proves tenen un pes del **30%** restant.