Sistemes Operatius II - Pràctica 2

Octubre del 2016

Índex

1 Introducció

L'objectiu d'aquesta pràctica és la preparació del codi perquè a la pràctica 4 es pugui aprofitar la capacitat multiprocessador de l'ordinador. Per això caldrà realitzar un codi modular i que segueixi unes determinades pautes indicades en aquest document. Cal esmentar que algunes de les pautes poden no tenir sentit en aquest moment, però no seguir-les implica que es tindrà més feina a la darrera pràctica.

Per implementar aquesta pràctica cal utilitzar el codi de la pràctica anterior. Assegureu-vos doncs que la pràctica 1 funciona correctament abans d'implementar aquesta pràctica.

2 La pràctica

A la figura 1 es mostra l'esquema de blocs a implementar. Observar que hi ha tres blocs i un algorisme que executa aquests tres blocs de forma iterativa.

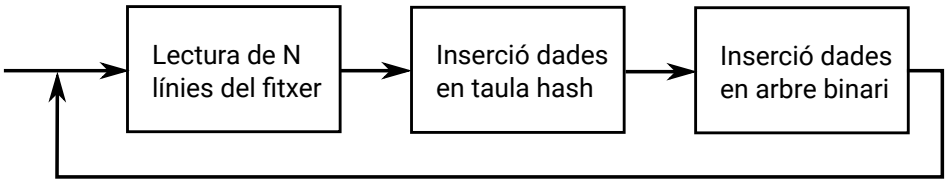

Figura 1: Esquema de blocs a implementar.

2.1 Lectura de N línies del fitxer

En aquesta pràctica les línies del fitxer a processar es llegiran en blocs de N línies. Observar que en cas que $N = 1$, s'obté una funcionalitat equivalent a la de la primera pràctica: les línies es llegeixen d'una en una. El bloc de N línies es passarà al següent procediment de l'esquema de blocs, el qual s'encarregarà d'extreure la informació de les columnes.

La raó de procedir a la lectura en blocs de N línies rau en el fet que a la pràctica 4 s'assignarà, a cada fil, un bloc de línies diferent. Cada fil s'executarà en un processador diferent i, per tant, cada processador s'encarregarà de processar un bloc de N línies diferent. Veurem que el valor d'N juga un paper important en l'eficiència de l'algorisme de la pràctica 4.

2.2 Inserció de les dades en una taula de hash

El bloc de N línies llegit pel bloc anterior és l'entrada a aquest bloc que s'encarregarà d'extreure, de cada línia, les columnes 4, 15, 17 i 18, fet a la pràctica anterior. A la pràctica 1 la la informació extreta s'insereix directament a l'arbre. En aquesta pràctica la informació extreta del bloc de N línies s'insereix "temporalment" en una taula de hash. Un cop s'hagi inserit tota la informació del bloc de N línies a la taula de hash es transferirà la informació a l'arbre. De nou, aquest "pas intermedi" de la taula de hash s'utilitza per facilitar la implementació multifil: a la pràctica 4 cada fil s'encarregarà de processar un bloc de N línies diferent i d'inserir la informació extreta en una taula de hash diferent. Els detalls els veurem a la pràctica 4.

A la figura 2 es mostra l'estructura de la taula de hash a implementar. La **taula de hash és un vector de llistes**, on la llista és la utilitzada a la pràctica 1. La mida de la taula de hash

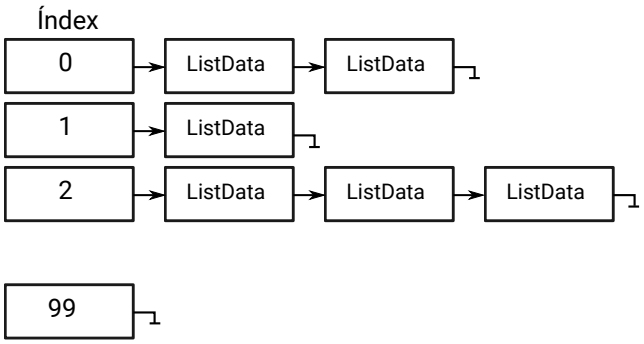

Figura 2: Taula de hash.

està parametritzada per un valor M; a la figura $M = 100$. L'origen de l'aeroport es farà servir per tal de trobar el valor de hash associat. Aquest valor de hash és l'índex a una llista enllaçada que emmagatzemarà la informació extreta per aquell aeroport d'origen. Veure la secció 3.1 per a detalls sobre la implementació.

2.3 Inserció de les dades a l'arbre binari

El darrer bloc tindrà com a entrada la taula de hash i s'encarregarà de bolcar la informació emmagatzemada a la taula de hash a l'arbre binari. A l'hora de bolcar la informació caldrà actualitzar tota la informació necessària. Veure la secció 3.3 per a detalls sobre la implementació.

3 Implementació i planificació

Es recomana que cada bloc de la figura 2 estigui implementat en una funció separada i que hi hagi una quarta funció que executi de forma iterativa els anteriors blocs. A continuació es donen alguns detalls i consells per a la implementació de les funcions.

3.1 Lectura de N línies del fitxer

El codi ha de tenir un paràmetre intern (mitjançant un `#define`, per exemple) que permeti fer experiments de forma senzilla amb diferents valors de N. És important poder fer experiments amb aquest paràmetre a la pràctica 4. Assegureu-vos que el codi funciona correctament amb valors d'N de, per exemple, 1000 o 10,000.

3.2 Inserció de les dades en una taula de hash

S'adjunta amb aquesta pràctica el codi `hash.c` com a exemple per calcular el valor de hash. El valor de hash és un sencer que va de 0 fins a $M - 1$. A la figura 2 el valor $M = 100$.

Com s'ha comentat abans, el valor de hash obtingut és un índex a una llista enllaçada, veure figura 2. Es recomana que es faci servir l'estructura `ListData` declarada a la pràctica 1. A mesura que s'extregui la informació del bloc de N línies, caldrà actualitzar les estructures `ListData` corresponents.

Caldrà, però, fer alguns canvis a l'estructura `ListData` per assegurar un funcionament correcte de la taula de hash. Observar que diferents aeroports poden mapar-se al mateix valor de hash. I

diferents aeroports poden tenir el mateix aeroport destí. Això pot provocar dificultats per poder manipular correctament les llistes.

Per solucionar aquesta dificultat es proposa ampliar `ListData` (de la pràctica 1) afegint un nou membre, `key_sec`, que estigui associat a una clau de cerca secundària. La clau principal, `key`, està associada a l'aeroport destí, mentre que `key_sec` estarà associat a l'aeroport d'origen. Caldrà definir una funció similar a `findNode` que permeti buscar en una llista fent servir dues claus, `key` i `key_sec`.

Mentre s'emmagatzema la informació a la taula de hash no es podrà accedir, en cap cas, a l'arbre binari per cercar-hi o actualitzar informació. Això és important per poder assegurar més endavant, a la pràctica 4, una bona implementació multifil.

3.3 Inserció de les dades a l'arbre binari

El darrer pas s'encarregarà de bolcar la informació de la taula de hash a l'arbre binari. Per facilitar aquest bolcament es recomana que el tipus `ListData` utilitzat per a l'arbre sigui el mateix que el tipus `ListData` de la taula de hash. A l'arbre no serà realment necessari, però, guardar la informació a `key_sec`. Es deixa a decisió de l'estudiant emmagatzemar-hi alguna informació.

Un cop s'hagi bolcat la informació de la taula de hash a l'arbre binari s'haurà d'alliberar tot allò necessari de la taula de hash perquè a la iteració següent es comenci amb una taula buida.

3.4 Els experiments

Per fer els experiments en aquesta pràctica es proposa utilitzar un fitxer CSV més gros que l'utilitzat a la primera pràctica. Mentre que a la pràctica 1 el fitxer a analitzar té 10,000 línies, es proposa que en aquesta pràctica el fitxer a analitzar tingui 100,000 línies. Inclús, si voleu, feu servir el fitxer complet que té més de 7,000,000 de línies. Per això

1. Aneu a la l'adreça http://stat-computing.org/dataexpo/2009/the-data.html i baixeu-vos el fitxer corresponent a l'any 2008. D'aquest darrer fitxer se us van proporcionar les 10,000 primeres línies a la pràctica 1.

2. El fitxer CSV baixat conté una capçalera d'una línia que, de moment, eliminarem manualment. Per eliminar la capçalera podeu utilitzar la instrucció `tail`. Així eviteu editar el fitxer.

3. Feu servir, per exemple, la instrucció `head` de la línia de comandes per extreure del fitxer tantes línies com us interessi analitzar. Se us proposa extreure 100,000 línies del fitxer.

Agafeu un nombre de línies suficientment gran per tal de provar diversos valors d'N, la mida del bloc de línies a llegir.

4 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 addicional en PDF. Aquí hi ha els detalls per cada directori:

- El directori `doc` ha de contenir un document (tres o quatre pàgines, màxim cinc pàgines, en format PDF, sense incloure la portada) explicant breument aquells detalls o decisions interessants de la vostra aplicació que no estiguin comentats a l'enunciat d'aquesta pràctica. No repetiu fil per randa el que es comenta en aquest enunciat i no cal que expliqueu tot el vostre codi.

- La carpeta `src` contindrà el codi font comentat (com a mínim les funcions). Els comentaris poden estar en anglès, català o castellà. 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.

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` amb servir el `fitxer.csv` (o un subconjunt d'aquest). El document té un pes del **20%** restant.