

Sistemes Operatius II - Pràctica 5

Desembre 2019

La cinquena pràctica se centra en la creació de l'arbre fent servir tècniques de concurrència. A la quarta pràctica es feien servir múltiples processos. En aquest cas es faran servir múltiples fils.

Índex

1 La pràctica	2
2 Implementació	2
2.1 Creació de l'arbre amb un únic fil	2
2.2 Creació de l'arbre fent servir múltiples fils	3
3 Entrega	3

1 La pràctica

Aquesta pràctica es crearà l'arbre fent servir tècniques concurrents. A la pràctica 4 la concurrència està implementada a nivell de procés i en aquesta pràctica es farà a nivell de fil. Es proposa que en aquesta pràctica la sincronització es realitzi a nivell de monitors (mutex).

Atès que en aquesta pràctica es faran servir fils secundaris per a la creació de l'arbre. Atès que els fils comparteixen l'espai de memòria del procés, no faran falta eines de comunicació interprocés tal com era necessari a la pràctica anterior. A grans trets, l'algorisme que es proposa implementar és el següent (es poden realitzar modificacions sobre aquesta proposta):

1. El fil principal llegeix el diccionari de paraules i crea l'estructura de l'arbre que anomenarem “arbre compartit”.
2. El fil principal crea un conjunt de fils secundaris que seran els encarregats de processar els fitxers de la base de dades.
3. Cada fil secundari tindrà una còpia pròpia de l'arbre. Anomenem-lo “arbre local”. Cada fil actualitzarà els valors dels nodes a mesura que es van llegint els fitxers de la base de dades. Tingueu en compte que cada fitxer només serà processat per un únic fil.
4. Quan un fil secundari no tingui més fitxers a processar actualitzarà l'arbre compartit a partir de l'arbre local que tingui.
5. El fil principal esperarà mentre els fils secundaris estiguin processant els fitxers de la base de dades. Quan tots els fils secundaris hagin acabat el fil principal tornarà a mostrar el menú.

Consulteu la secció d'implementació per tenir una proposta de com implementar el codi.

2 Implementació

Aquesta secció dona alguns consells per tal d'implementar correctament aquesta pràctica. Tingueu en compte que només caldrà modificar el codi de creació de l'arbre (i.e. l'opció 1 del menú). Es proposa seguir el següent ordre a l'hora d'implementar la pràctica. Sou lliures de fer altres implementacions i modificar el codi que se us proporciona a les vostres necessitats.

2.1 Creació de l'arbre amb un únic fil

Començarem la creació de l'arbre fent servir només un únic fil. L'objectiu és comprovar que podeu manipular fils. Observeu que encara no fa falta utilitzar sincronització. El procediment a implementar seria:

1. El fil principal llegeix el diccionari de paraules i crea l'estructura de l'arbre compartida.
2. El fil principal crea un únic fil secundari i li passa tota la informació que li fa falta per poder processar la base de dades.
3. El fil secundari crea una còpia pròpia de l'arbre i processa els fitxers de la base de dades.
4. Un cop s'han processat tots els fitxers, el fil secundari actualitzarà l'arbre compartit a partir de la informació de la còpia pròpia.

5. El fil principal espera mentre el fil secundari va processant la base de dades. Un cop el fil secundari ha acabat, el fil principal mostra per pantalla el menú.

Comproveu que l'arbre es crea correctament fent, per exemple, consultes als nodes de l'arbre. Assegureu-vos que valgrind no dona cap mena d'error.

2.2 Creació de l'arbre fent servir múltiples fils

A continuació es crearà l'arbre fent servir múltiples fils.

1. El fil principal llegeix el diccionari de paraules i crea l'estructura de l'arbre compartida.
2. El fil principal crea múltiples fils secundaris i els passa tota la informació que els fa falta per poder processar la base de dades.
3. Cada fil secundari crea la seva còpia local de l'arbre.
4. Els fils secundaris llegeixen de forma concurrent els noms de fitxers i actualitzen la seva pròpia còpia local de l'arbre. Assegureu-vos que cada fitxer de la base de dades només es processa per un únic fil secundari. Haureu de sincronitzar!
5. Quan un fil secundari no tingui més fitxers a processar haurà d'actualitzar l'arbre compartit a partir de la seva pròpia còpia local. De nou, haureu de sincronitzar!
6. El fil principal esperarà mentre els fils secundaris estiguin processant els fitxers de la base de dades. Quan tots els fils secundaris hagin acabat el fil principal tornarà a mostrar el menú.

Comproveu que, en consultar els nodes de l'arbre, us surten els mateixos resultats que amb l'aplicació sense fils.

3 Entrega

El fitxer que entregueu s'ha d'anomenar `P5_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 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:

- La carpeta `src` contindrà el codi font. S'hi han d'incloure tots els fitxers necessaris per compilar i generar l'executable. No cal incloure la base de dades ni el fitxer de diccionari. El codi ha de compilar sota Linux amb la instrucció `make`. És recomanable que poseu les funcions en fitxers diferents agrupats per la seva funcionalitat (per exemple, un fitxer per la creació de l'arbre, un altre per llegir i desar l'arbre, ...). El codi font ha d'estar comentat (com a mínim les funcions).

Per simplificar, es pot suposar que el codi s'executa dins del directori en què està situada la base de dades. El fitxer de diccionari (que pot tenir qualsevol nom) també estarà situat en aquest directori. El codi no tindrà cap paràmetre, atès que tot el control es farà des del menú de l'aplicació

`practica5`

- El directori `doc` ha de contenir un document (màxim tres pàgines, en format PDF, sense incloure la portada). Es demana respondre a dues preguntes
 1. Heu obtingut alguna millora en l'execució del codi respecte a fer servir només 1 fil? Podeu comparar-ho amb el que heu fet a la pràctica 4? Hi ha alguna diferència en el temps d'execució del codi? Comenteu les vostres respostes.
 2. Els fils secundaris, a l'hora d'actualitzar l'arbre compartit a partir de la seva pròpia local, pot utilitzar diverses estratègies: a) fer una secció crítica que bloquegi tot l'arbre mentre un fil fa l'actualització a partir de la seva pròpia local, b) fer una secció crítica que bloquegi cada node de forma independent cada cop que calgui actualitzar el comptador. Aquesta darrera opció permet que múltiples fils actualitzin al mateix temps l'arbre compartit a partir de la seva còpia local. Creieu que té alguna avantatge utilitzar l'opció b) respecte l'a)? Heu pogut fer proves per comparar les dues propostes? Sou lliures de mostrar altres solucions implementades.

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 es comprovarà el bon funcionament del vostre codi fent servir el `valgrind`. El document té un pes del **20%** restant.