

Sistemes Operatius II - Pràctica 4

Desembre del 2013

La quarta i darrera pràctica de sistemes operatius 2 se centra en aprendre a coordinar múltiples fils d'execució. La data d'entrega d'aquesta darrera pràctica és el diumenge **22 de desembre del 2013**.

Índex

1	Introducció	2
2	La pràctica	2
3	Implementació	3
4	Planificació	4
5	Entrega	4

1 Introducció

La tercera pràctica de sistemes operatius s'ha centrat en l'ús de múltiples fils per a la creació de l'arbre. En concret, recordem el procediment implementat a la pràctica 3:

1. En executar el vostre programa només hi ha un fil. Anomenem a aquest fil el fil principal. Aquest fil és el que imprimeix per pantalla el menú, el que permet desar l'arbre a disc, carregar-lo de disc o bé analitzar les paraules de l'arbre.
2. En seleccionar del menú l'opció de creació d'arbre el fil principal demana el fitxer de configuració. El fil principal obre aquest fitxer i llegeix només la primera línia, que conté el nombre de fitxers a processar. En aquest moment es creen un mínim de dos fils que analitzen els fitxers i creen l'arbre. El fil principal es queda esperant que els fils creats finalitzin la creació de l'arbre.
3. Un cop s'han processat tots els fitxers el fil principal es desperta i torna a visualitzar el menú.

En aquesta quarta i darrera pràctica es continuarà utilitzant múltiples fils, igual que a la pràctica 3, per a la creació de l'arbre però utilitzant una coordinació més complexa. A les següents seccions es detallen les tasques a realitzar.

2 La pràctica

En aquesta pràctica es modificarà la pràctica anterior per implementar la següent funcionalitat

1. En executar el programa només hi ha un fil d'execució. El fil principal crearà, abans de mostrar el menú per pantalla, tots els fils necessaris per crear l'arbre. Aquests fils els anomenarem fils secundaris. Els fils secundaris es posaran a dormir (ja que no cal crear cap arbre en aquest moment). El fil principal mostrarà el menú un cop tots els fils creats s'hagin posat a dormir.
2. En seleccionar del menú l'opció de creació d'arbre, el fil principal demanarà per teclat el fitxer de configuració. El fil principal obrirà aquest fitxer i llegirà només la primera línia, que conté el nombre de fitxers a processar. En aquest moment el fil principal desperta els fils secundaris. Els fils secundaris començaran a crear l'arbre igual que s'ha fet a la pràctica 3: cada fitxer es processa analitzant les paraules i copiant-les a una estructura local. Mentrestant el fil principal es posa a dormir.
3. En acabar la creació de l'arbre es desperta el fil principal i els fils secundaris es posen a dormir.

4. En sortir del programa el fil principal despertarà els fils secundaris per indicar-los que han de finalitzar. El fil principal no pot sortir del programa fins que tots els fils secundaris hagin finalitzat.

Observar que en aquesta pràctica els fils són creats a l'inici de programa i no finalitzen fins que se surt de l'aplicació. En aquesta pràctica caldrà doncs utilitzar les funcions `pthread_cond_wait`, `pthread_cond_signal` i `pthread_cond_broadcast` de la llibreria de `threads`.

3 Implementació

Per a la implementació de la solució es demana utilitzar monitors. A continuació es descriu l'esquema a implementar (cada grup de pràctiques tindrà variacions sobre aquesta proposta):

1. En executar el vostre programa, només hi haurà un fil. Anomenarem a aquest fil el fil principal. El fil principal crearà, abans de mostrar el menú, tots els fils secundaris necessaris. Els fils secundaris es posaran a dormir de seguida amb la funció `pthread_cond_wait`. Heu d'assegurar-vos que tots els fils creats s'han posat a dormir abans que el fil principal mostri el menú. Per això es proposa utilitzar un esquema similar a una barrera: el fil principal es posa a dormir a un `pthread_cond_wait` i es desperta així que el darrer fil es posi a dormir. Un cop s'hagi despertat el fil principal aquest mostrarà el menú per pantalla. No es poden utilitzar funcions similars a `sleep` per implementar aquesta funcionalitat.
2. En seleccionar del menú l'opció de creació d'arbre, el fil principal demanarà per teclat el fitxer de configuració. El fil principal obrirà aquest fitxer i llegirà només la primera línia, que conté el nombre de fitxers a processar. En aquest moment el fil principal ha de despertar els fils secundaris adormits fent servir, per exemple, el `pthread_cond_broadcast`. Haureu de passar a cada fil la informació necessària perquè puguin accedir als fitxers de la llista de configuració i puguin copiar les dades a l'arbre.
3. El fil principal esperarà a un `pthread_cond_wait` mentre els fils secundaris creen l'arbre. Els fils secundaris que finalitzen la creació de l'arbre es posaran a dormir de nou amb un `pthread_cond_wait`. El darrer fil que es posi a dormir serà l'encarregat de despertar el fil principal amb un `pthread_cond_signal`. Un cop s'hagi despertat el fil principal es tornarà a mostrar el menú principal per pantalla.
4. En seleccionar l'usuari l'opció per sortir del programa el fil principal haurà de despertar els fils secundaris. El fil principal haurà de notificar els fils secundaris que han de finalitzar mitjançant una variable. El fil principal utilitzarà la funció `pthread_join` per esperar que cadascun dels fils finalitzi. No es pot utilitzar la funció `pthread_cancel` per implementar aquesta part.

4 Planificació

Aquí hi ha unes recomanacions per planificar-vos la feina:

1. Per fer aquesta pràctica és important entendre bé el funcionament de les funcions `pthread_cond_wait` i `pthread_cond_broadcast`. Llegiu i experimenteu amb la segona part de la fitxa 4 penjada al campus que parla de les variables condicionals.
2. Implementeu el codi que permet crear els fils a l'inici de l'execució del programa i abans de mostrar el menú. El primer que hauran de fer el fils secundaris es posar-se a dormir. No us preocupeu, de moment, pel problema d'assegurar que els fils estiguin adormits abans de mostrar el menú. Això es farà al punt 5 d'aquesta llista.
3. En escollir l'usuari l'opció de crear l'arbre haureu de despertar els fils secundaris. Implementeu l'algorisme corresponent que permet que el fil principal es posi a dormir mentre els secundaris creen l'arbre. En finalitzar la creació de l'arbre, els fils secundaris es posaran a dormir un a un. El darrer fil que es posi a dormir haurà de despertar (abans de posar-se a dormir) el fil principal.
4. En escollir l'usuari l'opció de sortir haureu de despertar els fils secundaris i notificar-los que cal sortir. Implementeu l'algorisme que permet despertar els fils secundaris i notificar-los (mitjançant una variable) que cal finalitzar. El fil principal utilitzarà la funció `pthread_join` per esperar l'acabament de cadascun dels fils.
5. Un cop hagueu après a coordinar els fils amb `pthread_cond_wait` i `pthread_cond_broadcast`, implementeu el punt 1 de la secció d'implementació. Heu d'assegurar-vos que tots els fils secundaris estan adormits abans de mostrar el menú per pantalla. Per això utilitzeu les funcions `pthread_cond_wait` i `pthread_cond_signal`: el fil principal es posarà a dormir amb un `pthread_cond_wait` un cop hagi creat tots els fils. Els fils secundaris es posaran a dormir amb un `pthread_cond_wait`. Així que el darrer fil es posi a dormir caldrà despertar el fil principal amb un `pthread_cond_signal`.

5 Entrega

El fitxer que entregueu s'ha d'anomenar `P4_Cognom1Cognom2.tar.gz` (o `.zip`, o `.rar`, etc), on `Cognom1` és el cognom del primer component de la parella i `Cognom2` 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 tres carpetes: `src`, que contindrà el codi font, `proves`, que contindrà resultats d'execució 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 en anglès en un format similar al mostrat als fitxers `red-black-tree.c` i `linked-list.c`. Es necessari comentar com a mínim les funcions que hi ha al codi.
- La carpeta `proves` ha d'incloure qualsevol informació que considereu rellevant. Aquesta informació haurà d'estar comentada a la memòria.
- El directori `doc` ha de contenir un document (dues o tres pàgines, en format PDF) explicant el funcionament de l'aplicació, la discussió de les proves realitzades i els problemes obtinguts. És particularment interessant que comenteu sobre els següents punts
 - Una explicació de com s'ha implementat el punt 1 de la secció d'implementació. És a dir, com assegureu que els fils secundaris estan adormits un cop el fil principal hagi mostrat el menú per pantalla. Ajudeu-vos d'una figura descriptiva en cas necessari.
 - Una explicació de com s'ha implementat el punt 2 i 3 de la secció d'implementació. És a dir, com es coordinen els fils entre sí a l'hora de crear l'arbre? Ajudeu-vos d'una figura descriptiva en cas necessari.
 - Una explicació de com s'ha implementat el punt 4 de la secció d'implementació. És a dir, com coordineu els fils secundaris quan l'usuari vol sortir del programa? Ajudeu-vos d'una figura descriptiva en cas necessari.

En aquest document no s'han d'explicar en detall les funcions o variables utilitzades. Sí que es pot explicar (incloent gràfics) les estructures utilitzades.

La data límit d'entrega d'aquesta pràctica és el **22 de desembre del 2013**. El codi tindrà un pes d'un 70% (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.) i el document i les proves el 30% restant.