



Presentación

Objetivos

Esta práctica tiene como objetivo la sincronización de una aplicación concurrente multi-hilo utilizando los mecanismos de sincronización de java. Para su realización se pondrán en práctica los conceptos de sincronización presentados en la asignatura.

Presentación de la práctica

Hay que presentar los archivos con el código fuente realizado. Toda la codificación se hará exclusivamente en Java.



Enunciado de la práctica

Esta práctica tiene como base la aplicación concurrente de indexación y búsqueda mediante un Índice Invertido realizada en la segunda práctica de la asignatura. En esta segunda práctica nos vamos a centrar en dotar a la aplicación de los mecanismos de sincronización oportunos que permita una ejecución correcta, determinista y eficiente de la misma.

Sincronización

En esta versión se deberá implementar las siguientes sincronizaciones y modificaciones:

- **Clase Thread-safe.** Se tiene que lograr que la clase Indexing sea thread-safe en su conjunto, de forma que pueda ser utilizada dentro de una aplicación concurrente por varios hilos al mismo tiempo.
- **Evitar las condiciones de carrera.** Se tendrá que analizar todas las operaciones que pueden generar condiciones de carrera y implementar las secciones críticas que se consideren oportunas para garantizar un funcionamiento correcto y determinista de la aplicación.

En esta versión, los hilos deberán poder trabajar directamente con el Hash global mediante la utilización de mecanismos de sincronización. No se permite que cada hilo trabaje con su propio hash local.

- **Estadísticas.** La aplicación calculará una serie de estadísticas locales/globales:
 - Número de keys diferentes generados/procesados.
 - Número values (offsets) generados/procesados.
 - Número de bytes del fichero leídos/escritos/procesados.
 - Porcentaje de Progreso (dependerá del método).

Al final de la ejecución de la aplicación se mostrarán tanto las estadísticas finales de cada hilo, como las estadísticas globales.

- **Progreso.** Cada M% de progreso (5%, 10%, 15%,... 100%) la aplicación mostrará las estadísticas globales hasta ese momento. M se pasará como parámetro a la aplicación (por defecto, M=10%). Mientras se calculan y muestran las estadísticas globales, ningún hilo podrá trabajar.
- **Espera resultado final.** En esta versión no se permite la utilizar los join para esperar a que los hilos finalicen.



SISTEMAS CONCURRENTES Y PARALELOS

PRÁCTICA 4: Sincronización Aplicaciones Concurrentes (Java)

Al implementar las sincronizaciones se tienen que utilizar todos los mecanismos de sincronización que hemos visto en clase, es decir: *synchronized*, locks, variables de condición y semáforos.



Funciones involucradas.

En la práctica podéis utilizar las siguientes clases y métodos de java:

- Palabra clave *synchronized*
- *clase java.util.concurrent.locks.condition:*
 - *signal()*
 - *void signalAll()*
 - *void await()*
- *clase java.lang.Object:*
 - *void notify()*
 - *void notifyAll()*
 - *void wait()*
- *clase java.util.concurrent.Semaphore:*
 - *void acquire()*
 - *void acquire(int permits)*
 - *void release()*
 - *void release(int permits)*



Análisis prestaciones

Calcular el tiempo de ejecución de la versión concurrente sincronizada y compararlo con el de la versión concurrente sin sincronizar y con la versión secuencial. Discutir los resultados obtenidos.

Analizar también, cómo evoluciona el tiempo de ejecución de la nueva versión en función del número de hilos utilizados en la aplicación concurrente. Entregar un pequeño informe en donde se muestren (preferiblemente de forma gráfica) y expliquen los resultados obtenidos.

Indicar en el informe las características del hardware en donde habéis obtenido los resultados (especialmente el número de procesadores/núcleos).



Evaluación

Se utilizarán los siguientes criterios a la hora de evaluar la práctica, partiendo de una nota base de 10 para esta versión:

- Evaluación implementación sincronización:
 - Condiciones Carrera (-2.0p)
 - No es thread-safe (-1.0p)



- Espera Activa (-1.0p)
- Hash Local (-1.0p)
- Estadísticas (-1.0p)
- Progreso (-1.0p)
- Esperar resultado final (-0.5p)
- No utilizar synchronized (-1.0p)
- No utilizar Locks (-1.0p)
- No utilizar Variables de condición (-1.0p)
- No utilizar Semáforos (-1.0p)
- Evaluación del informe de la práctica:
 - + Informe completo, con gráficos y conclusiones correctas (+0.25p,+1.0p)
 - Sin informe (-1.5p)
 - Informe con incoherencias en los tiempos sin justificar (-0.5p)
 - Informe: análisis poco riguroso (-0.5p)
 - No se explica el diseño de la solución de sincronización propuesta (-0.5p)
 - No se describe los problemas encontrados y como se han solventado (-0.5p)
- Estilo de codificación:
 - El código no está correctamente comentado (-0.5p)
 - No utilizar una correcta tabulación y organización del código (-0.5p)
 - Sin descomposición funcional (el código no es modular, ni está estructurado) (-0.5p)
 - Sin control de errores (-0.5p)
 - + Test unitarios para verificar la correcta ejecución de la lógica de la aplicación (+0.5p)

Se puede tener en cuenta criterios adicionales en función de la implementación entregada.



Formato de entrega

MUY IMPORTANTE: La entrega de código que no compile correctamente, implicará suspender TODA la práctica.

No se aceptarán prácticas entregadas fuera de plazo (salvo por razones muy justificadas).

La entrega presencial de estos ejercicios es obligatoria.

Comenzar vuestros programas con los comentarios:

```
/* -----  
Práctica 4.  
Código fuente : gestor.c  
Grado Informática  
NIF y Nombre completo autor1.  
NIF y Nombre completo autor2.  
----- */
```



SISTEMAS CONCURRENTES Y PARALELOS

PRÁCTICA 4: Sincronización Aplicaciones Concurrentes (Java)

Para presentar la práctica dirigiros al apartado de Actividades del Campus Virtual de la asignatura de Sistemas Concurrentes y Paralelos, id a la actividad "Práctica 4" y seguid las instrucciones allí indicadas.

Se creará un fichero tar con todos los ficheros fuente de la práctica, con el siguiente comando:

```
$ tar -cvf prac4.tgz fichero1 fichero2 ...
```

se creará el fichero "prac4.tgz" en donde se habrán empaquetado y comprimido los ficheros "fichero1", fichero2, y ...

Para extraer la información de un fichero tar se puede utilizar el comando:

```
$ tar -xvf tot.tar
```

El nombre del fichero tar tendrá el siguiente formato: "Apellido1Apellido2PRA4.tar". Los apellidos se escribirán sin acentos. Si hay dos autores, indicar los dos apellidos de cada uno de los autores separados por "_". Por ejemplo, el estudiante "Perico Pirulo Palotes" utilizará el nombre de fichero: PiruloPalotesPRA4.tar



Fecha de entrega

Entrega a través de Sakai el 17 de diciembre a las 11h, entrega presencial la semana del 16 al 20 de diciembre de 2019.

