

Qüestionari sobre threads i concurrència:

1-. Un sistema de programació paral·lela és un sistema de múltiples processadors, els quals cada un té la seva memòria local, i que es poden comunicar els uns amb els altres a partir d'una memòria compartida (shared memory). És un sistema pensat per paral·lelitzar processos seqüencials i fer-los més ràpids, encara que no sempre s'aconsegueix.

Un exemple d'aquest sistema és un PC, ja que té 3 cores amb una cache cada una i, a més una RAM de memòria compartida.

Per altra banda, un sistema distribuït té varis processadors que estan connectats per xarxa. Per tant, la connexió entre processadors es realitza mitjançant sockets i la comunicació entre ells s'aconsegueix a partir de protocols de xarxa.

Els principals desavantatges d'un sistema de programació paral·lela són:

- *Escalabilitat*: al compartir memòria (shared memory) es converteix en un 'coll d'ampolla'.
- *Fiabilitat*: un fallo en un dels processadors comportarà la fallida del sistema.

Els principals desavantatges d'un sistema de programació distribuït són:

- *Replicació*: es necessita una velocitat molt ràpida de propagació de la informació, els quals també tenen possibilitats de fallo.
- *Seguretat*: es pot accedir al sistema des de diferents llocs fet que pot fer que s'intercepti tràfic en la xarxa.
- *Manteneabilitat*: Las diversas computadoras de un sistema pueden ser de diferentes tipos o ejecutar diferentes sistemas operativos. Los fallos en una máquina pueden propagarse entre unas i altres.

2-. El codi relacionat es troba en el projecte ParallelSystemEx2.

3-. El codi relacionat es troba en el projecte ParallelSystemEx3.

4-. El codi relacionat es troba en el projecte ParallelSystemEx4.

5-. La principal diferència és que quan un programa crida el mètode start() d'un thread, aquest crea una nou Thread i el codi que es troba dins del mètode run() s'executa. En canvi, cridant directament al mètode run() del thread el codi de dins d'aquesta funció s'executa igualment però dins del Thread actual que l'ha cridat, no es crea un nou fil d'execució.

Una altra diferència és que no hi ha possibilitat de cridar dues vegades el mètode start() en un mateix objecte de thread, en canvi sí que es possible fer-ho amb el mètode run().

6-. El codi relacionat es troba en el projecte ParallelSystemEx6.

7-. Executats els dos codis, s'ha comprovat que la diferència entre les dues execucions només es suficientment efectiva quan el número d'elements a ordenar és molt gran i l'array tendeix a estar casi del tot ordenat, en cas contrari sinó, la diferència no es del tot important ja que la gestió de threads també aporta un delay en el rendiment de l'aplicació. A més a més, executar tants threads paral·lels pot provocar que moltes màquines no ho suportin i problemes en la sincronització de tots ells.

El codi que s'ha utilitzat per realitzar la cerca seqüencial ha sigut el següent:

```
public static void SequentialOrder(int array[]) {  
    int size = array.length;  
    for ( int i=0; i<size-1; i++) {  
        for (int j=(i+1); j<size; j++) {  
            if ( array[j] < array[i] ) {  
                int tmp = array[i];  
                array[i] = array[j];  
                array[j] = tmp;  
            }  
        }  
    }  
}
```