**Esercitazione #1 di Sistemi Operativi**

*Multithreading in Java*

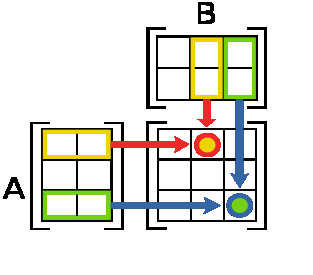
*Patrizia Scandurra – a.a. 2016-17*

1. **Esercizio**: creare un’applicazione Java contenente 2 thread: un thread principale (main()) crea e avvia un nuovo thread, e poi ogni 500 ms controlla se il figlio è ancora vivo. Se il figlio ha terminato, termina anche lui. Il processo padre deve anche produrre varie stampe a video per descrivere l’attività, ed in particolare deve stampare il numero di controlli che effettua. Il thread figlio effettua per *n* volte una generica attività *A* di una certa durata *s* (usare il metodo sleep per simulare un’attivita di *s* secondi). Provare ad eseguire l’applicazione. Cosa succede? Provate a variare la durata di *A* ed il parametro *n*, ed eseguite di nuovo.
2. **Esercizio (MatrixSum parallela).** Si vogliono sommare tutti gli elementi di una matrice intera NxM con un insieme N di thread (un thread per riga), ciascuno che somma i numeri di una riga. Il thread principale deve sincronizzarsi sulla terminazione dei thread figli (*join multiplo*) e stampare poi il risultato finale (la somma dei totali delle righe) a video.
3. **Esercizio (Fibonacci)** Scrivere un programma multithread in Java che generi la successione di Fibonacci in base ad una certa quantità N di numeri specificati dall’utente. La successione di Fibonacci è una successione di numeri interi naturali definibile assegnando i valori dei due primi termini, F0:= 0 ed F1:= 1, e chiedendo che per ogni successivo n>1 sia **Fn = Fn-1 + Fn-2**

Ad es. per N=10, la successione di numeri di Fibonacci è: 0,1,1,2,3,5,8,13,21,34.

Il programma deve creare un thread separato che scrive i numeri di Fibonacci.

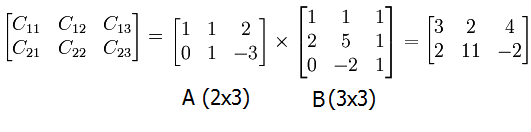
1. **Esercizio (Prodotto di Matrici)**. Scrivere un programma multithread in Java che calcola il prodotto C (m×p) tra due matrici A (m×n) e B(n×p) così definito:

per ogni i=1..m e j=1..p.

Il programma deve calcolare ogni elemento C(i,j) in un thread separato (ciò comporta la creazione di m×p

thread). Testare il programma con il seguente esempio:



*[****Suggerimento****: Usare due cicli (doppi con i,j) separati: uno per lo start di mxp thread, ed uno per il join su mxp thread t(i,j).]*