Ukeoppgaver uke2 i INF2440 - v2015

1. Avslutning av parallelle tråder.

Vi skal ta utgangspunkt i oppgave 2 i uke1 – finne det største elementet i en array av lengde n med tilfeldig innhold – trukket fra nextInt() i klassen Random. I denne oppgaven skal vi bruke n = 1000, 100 000 og 10 mill. Når du lager parallelle versjoner av denne algoritmen, skal du bruke den idéen at hver tråd har en lokal variabel '**`int minMax`**' og søker i sin del av hele arrayen slik at tråd 0 har a[0..n/k], tråd 1 har a[n/k+1..2*n/k] ,…osv..

Når en tråd har funnet max i sin del av arrayen, kaller de på en metode: synchronized **`void finnGlobalMax(int minMax) {..},`** som finner max i en felles, global variabel '**`int globalMax`**':

a) Hvis du ikke har programmert den sekvensielle løsningen, så gjør det.
b) Vi skal nå lage tre parallelle versjoner med k tråder (k= antall kjerner du har på maskinen) og så teste ut de 3 måtene vi har lært for å avslutte parallelle tråder:
   b.1) med array av tråder og kall på join() metoden i main tråden på hver av disse.
   b.2) med en CyclicBarrier som brukes av både main-tråden og de k andre trådene.
   b.3) med en Semaphore som også brukes av de k trådene.

   Skriv de tre måtene i samme program, og ta tidene én gang på hver av de fire delprogrammene – ett sekvensielt og de tre parallelle.
   Kan du si noe om hvilken måte å avslutte på som er raskest?

c) Implementer den løkka som gjør disse fire tidsmålingene f.eks 9 ganger i samme programmet. Du skriver de observerte tidene for hver av de fire metodene ned i en double-array. Når du er ferdig med tidtakingen, sorterer du arrayene med innstikksortering og skriver ut for hver av metodene medianen du da finner i tids-arrayene på plass: tidArrayX[(array.length-1)/2].

Skriv en kort rapport med disse tidene og om du synes du kan konkludere med hvilken metode er raskest/enklest/best til å avslutte trådene.

2. Jobb med Oblig1.

Ta utgangspukt i hvordan du løser FinnMaks problemet og den måten du fant var raskest å dele opp data for den parallelle løsningen, og overfør den type oppdeling til «FinnDe50Største» problemet.