

INF1010 - Obligatorisk oppgave 6, vår 2014

Versjon 0.9

Oppgavebeskrivelse

Du skal lage et program som sorterer ord ved hjelp av flere tråder. Programmet skal ta imot tre parametre på kommandolinjen: et heltall som angir antall tråder som skal brukes til sortering, og filnavn til en innfil og en utfil. Eksempel på kommandolinje

```
$ java Sort 128 names.txt out.txt
```

I dette eksemplet skal programmet bruke 128 tråder, lese inn ord fra filen "names.txt", og skrive resultatet ut til filen "out.txt". Første parameter i kommandolinjen kaller vi «antTråden» («threadCnt») og angir antall tråder som skal brukes til sorteringen. Første linje i innfilen er et heltall som vi kaller «antOrd» («wordCnt») og angir hvor mange ord det er totalt i resten av filen. Resten av filen inneholder ordene som skal sorteres, adskilt med et eller flere linjeskift

Programmet skal lese ordene inn i en tabell (array). Hver tråd skal sortere $N = \text{«antOrd»} / \text{«antTråden»}$ ord (+/- 1). Les først alle ordene inn i en tabell i minnet og starter deretter alle trådene. På denne måten vil du kunne se at flere tråder sorterer raskere enn en (eller noen få). Du skal selv programmere (og skjønne alle detaljer i) den sorteringsalgoritmen du velger. Lag gjerne en enkel algoritme med en grei invariant. Beskriv invarianten du bruker som en (eller flere) kommentarer i programmet ditt.

Når minst to tråder er ferdig med å sortere sine deler av de innleste ordene skal en tråd (du kan velge om det skal være en ny tråd eller en av trådene som allerede kjører) flette (se nedenfor om hva fletting er) sammen de ferdig sorterte delene to og to, osv. inntil alle ord fra innfilen er ferdig sortert. Lag så mange tråder at flettingeneskjer i parallell og dermed hurtig hvis maskinen har nok prosessorer/kjerner. (Legg merke til at det etter hvert er langt færre enn «antTråden» tråder som jobber med å flette).

Du skal også levere en tekstfil der du skriver et lite og uformelt argument om:

- 1. Parallellitet** Hvilke operasjoner kan gå i parallell og hvilke kan ikke gå i parallell i programmet ditt (Amdahls lov)?
- 2. Kjøretiden** programmet ditt bruker på sorteringen i forhold til «antOrd» og «antTråden», f.eks. ca. hvor mye øker sorterings-kjøretiden (eller antall operasjoner som utføres) når «antOrd» dobler seg. Stor O-notasjon er ikke pensum og trenger derfor ikke tas med; et lite anslag ved dobling av dataene (og dobling av antall prosessorer) er nok.

Programmet skal i alle fall virke med «antTråden» mellom 1 og 1000 (og gjerne flere), og skal bl.a. kunne sortere de to datafilene vist nedenfor. Hvis antall ord i innfilen ikke stemmer med heltallet på første linje skal programmet stoppe med en fornuftig feilmelding. Før programmet skriver det sorterte resultatet på utfilen skal alle ord ligge i en tabell (array), og programmet skal sjekke at denne tabellen har riktig lengde («antOrd») og at det ikke ligger en null-peker som siste element

Om fletting

Når vi fletter to sorterte sekvenser, så ser vi på en måte for oss at de ligger ved siden av hverandre og at vi så "trer" dem inn i hverandre, akkurat som vi gjør når vi fletter tau (men akkurat slik er vanskelig/umulig/ineffektivt når vi skal sortere to tabeller). Om vi skal flette to tabeller (arrayer) så legger vi resultatet i en ny tabell. Her er et eksempel: Ferdig sortert 1: 1, 7, 12, 23, 29. Ferdig sortert 2: 2, 19, 23, 31, 33, 51. Resultat av å flette 1 og 2: 1, 2, 7, 12, 19, 23, 23, 29, 31, 33, 51.

Datafiler

Du kan bruke disse filene til testing. Ikke ta dem med i innleveringen din:

[names.txt](#) 5163 ord, 40 KB Fra [census.gov](#)

[sowpods.txt](#) 267751 ord, 2905 KB Fra [isc.ro](#)

Arbeid og innlevering

Dette er en relativt kort oppgave som det er ment du skal jobbe med fra 30. april til 13. mai 2014. Du kan selvfølgelig begynne på oppgaven når du vil; på forelesningen onsdag 9. april gikk vi gjennom mye av det du trenger, men først på forelesningen onsdag 23. april har vi vært gjennom alt stoffet du trenger til denne oppgaven.

Dette er en individuell obligatorisk oppgave. Det bør snakke med andre studenter om hvordan dere skal løse oppgaven, men du skal likevel skrive ditt eget program, du skal selv taste inn alle tegn i programmet og du skal skjønne alle deler av det. Du skal ikke skrive av koden til en annen. Hvis du ikke kan gjøre rede for alle deler av programmet vil besvarelsen din ikke bli godkjent. Hvis du har samarbeidet med noen om en algoritme eller måte å løse oppgaven på, så bør du nevne dette i besvarelsen din.

Siste frist for å levere obligen er **TIRSDAG 13. mai kl. 23:59**. Denne gangen er innleveringsfristen mer absolutt enn tidligere fordi det er lite tid til retting og tilbakemelding før eksamen. Det blir svært begrenset anledning til å levere revidert utgave etter fristen. **Ta kontakt med retteren din på forhånd hvis du tror at programmet du leverer kanskje ikke blir godkjent.**

I tillegg til Java-programmet skal du levere en tekstfil kalt "TIL_RETTER.txt" der du svarer på spørsmålene om kjøretid og parallelitet nevnt ovenfor, og skriver litt info til retter om besvarelsen din, bl.a. om evt. mangler i besvarelsen eller andre spørsmål du har. Hvis alt fungerer skriver du det. Ord-filer og .class-filer skal IKKE

være med i leveringen. Hovedprogrammet skal hete Sort.java og det skal gå an å kjøre det som vist i eksemplet ovenfor. Kontakt retteren din på forhånd hvis du vil gjøre det på en annen måte (og gjør det bare på en annen måte hvis du har fått skriftelig godkjenning på epost).