

INF1010 - Innleveringsoppgave 9

Frist: Onsdag 20. april, 10:00

Maks 8 poeng

Temaet for ukene er Tråder og parallellprogrammering (simultant med parprogrammering).

Oppgaven

Det skal lages et program som sorterer ord ved hjelp av 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 eksempelet over skal javaprogrammet Sort bruke 128 tråder, lese inn ord fra filen "names.txt", og skrive resultatet ut til filen "out.txt".

Første parameter i kommandolinjen kaller vi `antTraader` ("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 alle ordene som skal sorteres, adskilt med linjeskift.

Programmet skal gi en passende feilmelding hvis kommandolinjeparameterene ikke stemmer (både manglende og feil type som f.eks tekst istedenfor et tall på `antTraader` posisjonen). Bruk gjerne try/catch for å få til dette.

Programmets krav

- Programmet skal lese ordene inn i en tabell (`array[]`). Hver tråd skal sortere $n = \text{antOrd} / \text{antTraader}$ ord (± 1). Pass på at tabellen er laget før trådene startes. På denne måten vil man oppleve at flere tråder sorterer raskere enn en (eller noen få) tråd(er). Sorteringsalgoritmen som blir valgt skal programmeres og forstås. Lag gjerne en enkel algoritme med en grei invariant. Beskriv invarianten dere bruker som en (eller flere) kommentarer i programmet ditt.

- Når en tråd er ferdig med å sortere sin del, skal den gi fra seg denne til en 'Monitor'.
- 'Monitoren' skal derfor kunne ta imot *en ferdig sortert* array. I tillegg skal 'Monitoren' også kunne gi fra seg *to ferdig sorterte* arrayer. Dette er fordi en tråd som enten har gitt fra seg den første sorterte arrayen eller en ferdigflettet array, skal forsøke å hente to nye arrayer fra 'Monitoren' for å flette dem sammen.
- Henting og fletting av to arrayer skal altså skje hver gang det er "to og to tråder som er ferdig"(dvs. *minst* to arrayer er klare i 'Monitoren'), inntil alle ord fra innfilen er ferdig sortert. Legg merke til at det etter hvert er langt færre enn 'antTraader' tråder som jobber med å flette samtidig.
- Programmet skal i alle fall virke med antTraader mellom 1 og 1000 (og gjerne flere), og skal bl.a. kunne sortere de tre datafilen lenket til under. 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:
 - tabellen har riktig lengde (ant0rd)
 - det ikke ligger en null-peker som siste element
 - tabellen er sortert
- Det skal i README-filen skrives et lite og uformelt argument om spørsmålene under, og kan bli spurt om dette under retting:
 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 ant0rd og antTraader, f.eks. ca. hvor mye øker sorterings-kjøretiden (eller antall operasjoner som utføres) når ant0rd dobler seg. Stor O-notasjon er ikke pensum og trenger derfor ikke tas med (for de som er kjent med det); et lite anslag ved dobling av dataene (og dobling av antall prosessorer) er nok.

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:

Tabell1
Echo
November
Oscar

Tabell2
Fox
Mike
Sierra

Fletting av de to tabellene blir:

Nytabell
Echo
Fox
Mike
November
Oscar
Sierra

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
linuxwords.txt	479623	5303 KB	Fra /usr/share/dict/linux.words på Ifi-filsystemet

Arbeid og innlevering

I tillegg til Java-programmet må dere huske å skrive README'en der dere svarer på spørsmålene om kjøretid og parallelitet nevnt ovenfor. Skriv også litt info til retter om besvarelsen din, bl.a. om evt. mangler i besvarelsen eller andre spørsmål dere har. Hvis alt fungerer skriver du det.

Ord-filer og .class-filer skal *ikke* være med i leveringen.

Hovedprogrammet skal gå an å kjøre som vist i eksemplet ovenfor, med passende feilmeldinger ved feil bruk.

Krav til innleveringen

1. Klassenavnet og filnavnet skal være identisk.
2. Klassenavn skal skrives med stor forbokstav.
3. Variabelnavn skal ha liten forbokstav.
4. Oppgaven må kunne kompilere og kjøre på IFI sine maskiner.
5. .class filer skal *ikke* leveres.
6. Ikke bruk æ, ø eller å i .java-filene(heller ikke som kommentarer eller utskrift).
7. Filene skal inneholde gode kommentarer som forklarer hva programmet gjør.
8. Programmet skal inneholde gode utskriftssetninger som gjør det enkelt for bruker å forstå.
9. Metodenavn skal skrives med liten forbokstav.
10. Koden skal være riktig indendert. Er du usikker, se Appendix J i Big Java.
11. Hver klasse skal ligge i sin egen .java-fil.

Fremgangsmåte for innleveringer i INF1010

1. Lag en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
 - 1) Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
 - 2) Hvor lang tid (ca) brukte du på innleveringen?
 - 3) Samarbeidet du med noen under innleveringen? Hvis ja, skriv brukernavn på den/de du samarbeidet med.
 - 4) Var det noen oppgaver du ikke fikk til? Hvis ja:
 - a) Hvilke(n) oppgave er det som ikke fungerer i innleveringen?
 - b) Hvorfor tror du at oppgaven ikke fungerer?
 - c) Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
2. Logg inn på Devilry.
3. Lever .java-filene og README.txt i *samme innlevering*.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.