

Operativsystemer og C

Ugeseddel 4 (uge 38)

Forelæsning: Fredag d. 19. september**Emner**

Tråde og deres fordele. Multitcore programmering. Matrix multiplikation med tråde. Brugetråde og kernetråde. Mange-til-en, en-til-en, mange-til-mange. Pthreads. Cancellation. Trådpuljer. Linux tråde og `clone()`. Trådschedulering. Dynamisk allokering af hukommelse i C.

Litteratur og andet kursusmateriale

Silberschatz, kapitel 4: afsnit 4.1-4.7 (ikke 4.4.2, 4.4.3, 4.7.1), kapitel 5, afsnit 5.4, 5.5.4.5

Øvelser: Fredag d. 19. september**Forberedelser**

*** Læs hele opgaveteksten igennem så du ved hvad du skal til øvelserne. ***

Til øvelserne i denne uge skal du bruge

- Adgang til en multicore Linux maskine.
- C compiler gcc og en tekst editor.

Opgave 1. Multitrådet sum (1 time)

Målet med denne opgave er, at du kan

- anvende tråde til at samarbejde om beregninger på en multicore processorer.

På side 161 i Silberschatzopgave vises koden til et C program, der bruger Pthreads API til at starte tråd som beregner sum-funktionen:

$$sum = \sum_{i=0}^N i$$

Dette program illustrerer anvendelsen af Pthreads men vil derudover ikke have nogen mærkbar effekt for brugeren af programmet.

I denne opgave skal du omskrive programmet sådan at det rent faktisk kører hurtigere (på en multicore maskine) end det ellers ville have gjort uden brug af tråde.

Hints:

- Du skal starte flere tråde (1-2-4-8) til at samarbejde om beregningsarbejdet.
- Du skal lave en **struct** med parametre til trådene, der fortæller hvilket arbejde tråden skal lave og efterfølgende gemmer på resultatet af trådens del-sum.
- Du skal desuden ændre fra at lave summen for integers (typen **double**) til at lave dem i flydende tal (typen **int**). Dette er nødvendigt for at summen ikke giver overflow ved store N .
- Du kan med fordel antage, at N divideret med antallet af tråde er et heltal.
- Husk at sikre at din CPU har flere cores (**cat /proc/cpuinfo**). Du kan også logge ind på **ssh.itu.dk** når du tester.