



# IN1010 - Seminar 9

- Tråder
- 

# Praktisk

Husk å sjekke emnesiden regelmessig

Undervisningstilbud:

- <https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/undervisningstilbud/>
- Jeg har konkrete spørsmål/problemer med min kode -> Labtime!
- Jeg vil ha mer liveprogrammering -> Plenumstime!
- Jeg vil jobbe med andre (og kanskje en kjapp recap av forelesning) -> Gruppetime!
- Jeg vil ha en recap av de vanskeligste konseptene fra forelesning -> Repetisjonsgruppe!



# Oppmøteregistrering:

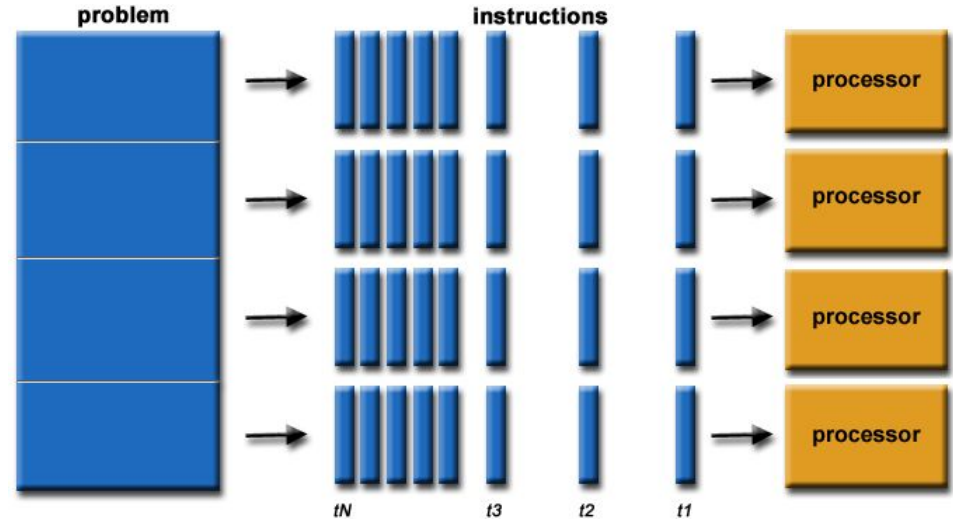
<https://nettskjema.no/a/188717>



Repetisjon fra denne uka

# Hva er et parallelt program og hvorfor bruker vi det

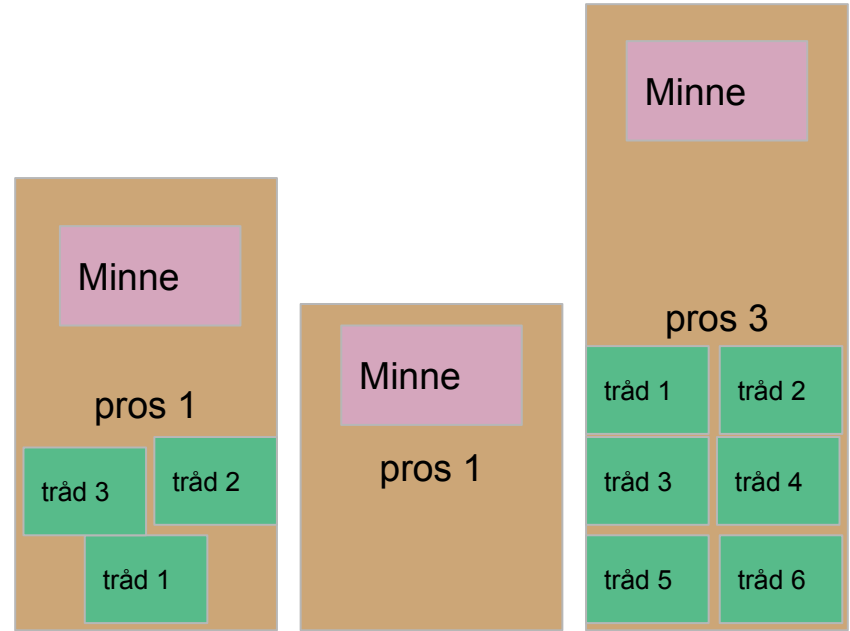
- CPU/ Kjerner
- Flere programmer for kjørt
- Maskinen har kapasiteten
- Dele opp problemet
- Raskere, men
  - Det tar tid å starte opp en tråd
  - Synkronisering



Bildekilde: [https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)

# Hva er en tråd?

- En parallell eksekvering inne i en prosess
  - En prosess er utføringen av et program
- Deler minne til prosessen
- Tråder kan gi i ekte parallel
- "Små"-prosesser i en vanlig prosess
- En sekvens med instruksjoner



# Hvordan lage en tråd ??

- Runnable steg 1

- <https://docs.oracle.com/javase/7/docs/api/java/lang/Runnable.html>

- Et interface

- run()

- Når du bruker .start() på en tråd vil objektets run metode bli kalt

- Thread

- [https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html#run\(\)](https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html#run())

- En klasse

- start()

- Tar inn vår Runnable klasse i konstruktøren

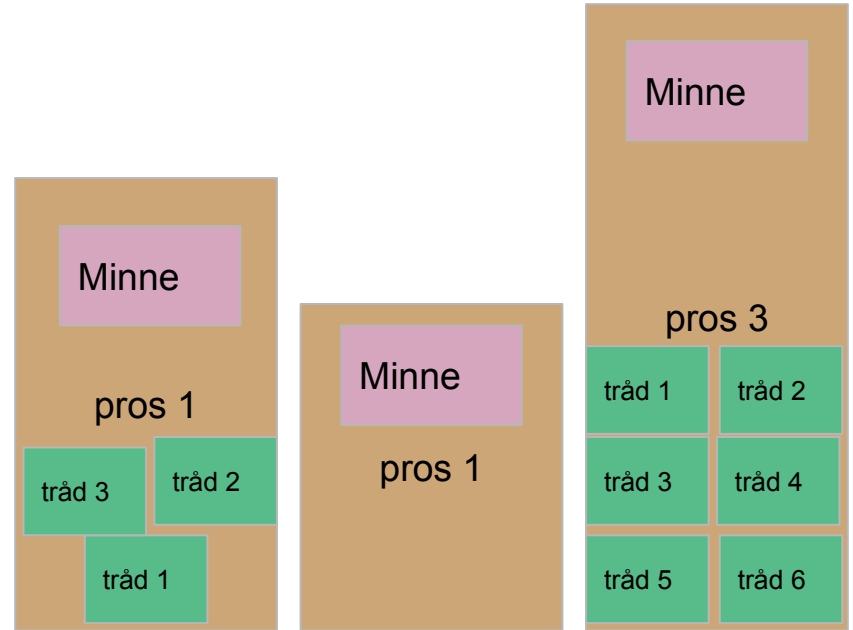
# Live-programmering

Lag et program som starter opp 3 tråder som printer alle tallene fra 0 til og med 9. Trådene skal ta inn en id som skal være en int.



# Delt data

- Data flere tråder deler
- Prosessens data
- Skrivning
- Lesing
- Lesing og skrivning
- Er du usikker lås



# Locks

- For å bruke en Lock må man importere
  - `java.util.concurrent.locks.Lock` (Et grensesnitt, bl.a. to metoder `lock()` og `unlock()` )
  - `java.util.concurrent.locks.ReentrantLock` (Lager de faktiske låseobjekten)
- Passer på at kun en tråd kan utføre koden om gangen
- Når man bruker Lock med `lock()` og `unlock()` er det viktig at man bruker en try-catch blokk. Nå må man også huske finally.
  - Finally er en kodesnutt som vil skje uansett. Når det kommer til Locks må man passe på å alltid `unlock()` i en finally blokk slik at hvis det skulle skje noe med en tråd kan de andre trådene få fortsette

```
46     public void metode(){
47         lock.lock()
48         try{
49             <kode her>
50         }catch(Exception e){
51             <hvis det er en exception og catche>
52         }finally{
53             lock.unlock() //vil alltid skje
54         }
55     }
```

# Monitor

- Et objekt som innkapsler den delte dataen
- Fungere som en type beskytter av felles data
- Monitor objektet har metoder som gjør at de andre klassen f.eks. kan endre data og hente ut data

PS: Den metoden blir brukt i mye i 1010 fordi det er en god objektorientert måte å gjøre synkronisering på

# Live-koding

Vi skal kode et program som starter opp 3 tråder (disse skal ta inn et tall i konstruktøren). Alle trådene skal prøve å legge til sitt tall i et felles array som ligger inne i en monitor. Det skal signaliseres tydelig til trådene når det ikke lenger er plass til flere tall i arrayet. Videre skal også tråden printe en liten beskjed som inneholder tallet sitt hver gang de får lag til tallet i arrayet.

# Condition

- En form for passiv venting
  - Vil bruke mindre ressurser enn aktiv venting
- `await()`: Får den tråden man er inne i nå til å vente til Condition gir signaliserer at det er greit å fortsette. (Eller at den blir interrupted)
- `signal()` : Signaliserer til en tråd om at den kan fortsett. (Som står å venter ved `await()`)
- `signalAll()`: Signaliserer til alle ventene tråder

# Ris, ros, forslag ?

<https://nettskjema.no/a/180345>

# Oppmøteregistrering:

<https://nettskjema.no/a/188717>

# Send meg en direkte melding i chatten

Vil du jobbe sammen med noen andre ? (ja /nei)

Hvis du har noen ønsker på hvem du vil jobbe med, så send det i samme melding

Svar gjerne også om svaret skulle være nei



# Breakoutrooms

1. Slå på kamera og ha en presentasjonsrunde
2. Jobb sammen med ukesoppgavene, de ligger på emnesiden -> grupper
  - a. Enten ved at én deler skjerm eller med [codecollab.io/](https://codecollab.io/)
  - b. OBS: codecollab er gratisjeneseter som UiO ikke har avtale med, sannsynligvis vil de samle data om dere. Dere kan fint løse oppgavene uten å bruke den tjenesten!
3. Bruk "ask for help"-knappen for å få hjelp 😊
4. Vi møtes her igjen for å gå gjennom oppgavene til slutt (dere bestemmer hvilke)

Jobbe med oppgaver