

DAT 103

Datamaskiner og operativsystemer (Computers and Operating Systems)

Øvelse 3 – Prosesser/tråder, prosessplanlegging og synkronisering – Hele øvelsen er obligatorisk

Dette er en øvelse hvor alle oppgavene er obligatoriske med innlevering. Hjelp til øvingen gis på laben i uke 44 – 46. Krav til den obligatoriske øvelsen:

- Dere kan arbeide sammen i grupper på inntil 4 personer.
- Gruppen leverer samlet på Canvas.
- Innleveringsfrist er **23:59, 13. november**.
- Dere må løse alle oppgavene.

Innhold

Oppgave 1	2
Oppgave 2 – process scheduling	2
Oppgave 3 – Process Synchronisation	2
Kritisk Region og semafor	2
Bounded Buffer	2
Readers/Writers	2
Dining Philosophers	3

Oppgave 1

1. Hva er et operativsystem?
2. Hva er en prosess (process)?
3. Hva er forskjellen mellom en tråd (thread) og en prosess (process)?
4. Ta utgangspunkt i en prosess som utfører koden nedenfor. Hvor mange barneprosesser vil bli opprettet?

```
fork();  
fork();  
fork();  
fork();
```

Oppgave 2 – Prosessplanlegging

1. Gitt følgende tabell for ankomsttiden (arrival time) og utbruddstiden (arrival time) til tre prosesser P_1 , P_2 and P_3 :

<i>Prosess</i>	<i>Ankomsttiden</i>	<i>Utbruddstiden</i>
P_1	0	9
P_2	1	4
P_3	2	9

Shortest-remaining-time-first, i.e., preemptive shortest job first, planleggingsalgoritme (scheduling algorithm) blir brukt. Planlegging utføres kun ved ankomst eller ferdigstillelse av prosesser.

- (a) Tegn Gantt-diagrammet for utførelsen.
 - (b) Hva er gjennomsnittlig ventetid for de tre prosessene?
2. Gitt følgende tabell for prosesseringstid (processing time) og periode til to prosesser P_1 og P_2 :

<i>Prosess</i>	<i>Prosesseringstid</i>	<i>Periode</i>
P_1	5	8
P_2	3	5

Fullførelsestidsfristene (The completion deadlines) for hver prosess er begynnelsen på dens neste perioden. For eksempel, tidsfristene for P_1 er ved tiden 8, 16, 24, ..., og tidsfristene for P_2 er ved tiden 5, 10, 15, 20, Bruk *Earliest Deadline First* for å planlegge de to prosessene. Ved *tiden* = 20, kan de to prosessene oppfylle alle tidsfristene? Tegn Gantt-diagram for utførelsen for å rettferdiggjøre svaret ditt. **Hint:** Earliest Deadline First er avbrytbar prioritetsbasert planlegging (priority scheduling with preemption).

Oppgave 3 – Process synkronisering

1. Kritisk Region og semafor
 - (a) Hva er en kritisk region (critical section)?
 - (b) Hva er en semafor?
 - (c) Hvordan kan vi implementere en kritisk region ved semaforer (semaphores)? Skriv en liten algoritme enten i Java, eller på norsk som en algoritme. Kall den kritiske regionen for "kritisk region" uansett algoritmespråk. Anta også at du har datatypen (klassen) Semaphore tilgjengelig med metodene wait og signal.

2. Readers/Writers

Lag en løsning i Java på problemstillingen “readers/writers”. Løs den først med semaforer slik boken har gjort det. Bytt deretter ut dine semaforer med Java sin Semaphore.

3. Dining Philosophers

Lag en løsning i Java på problemstillingen “the dining-philosophers”. Løs den først med semaforer slik boken har gjort det. Bytt deretter ut dine semaforer med Java sin Semaphore.