

Øvelse 4 – Synkronisering av prosesser/tråder – Hele øvelsen er obligatorisk

Dette er en øvelse hvor alle oppgavene er obligatoriske med innlevering. Hjelp til øvingen gis på laben i ukene 45 til 47.

Krav til den obligatoriske øvelsen:

- Dere kan arbeide sammen i grupper på inntil 4 personer.
- Gruppen leverer samlet på It's Learning.
- Innleveringsfrist er 20. november.
- Dere må løse alle oppgavene.

Innhold

Oppgave 1	2
Oppgave 2	2
Oppgave 3	2
Oppgave 4	2

Oppgave 1

- Hva er et operativsystem?
- Hva er en prosess?
- Hva er forskjellen mellom en tråd og en prosess?
- Hva er en kritisk region (critical section)?
- Hva er en semafor?
- Hvordan kan vi implementere en kritisk region ved semaforer? 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.
- Gi en kort beskrivelse av “banksjefens algoritme”.
- Vi skal se på 4 prosesser P₁, P₂, P₃, P₄ og deres bruk av ressurstypene A, B og C. Vi har følgende tilstand:

	Allokert			Maks behov			Tilgjengelig		
	A	B	C	A	B	C	A	B	C
							2	3	0
P ₁	0	1	0	7	5	3			
P ₂	3	0	2	3	2	2			
P ₃	3	0	2	9	0	2			
P ₄	1	1	2	4	3	3			

- Vis at systemet er i en sikker tilstand.
- Kan forespørselen (2 1 0) fra P₄ godkjennes? Begrunn!
- Kan forespørselen (1 1 0) fra P₁ godkjennes? Begrunn!

Oppgave 2

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

Oppgave 3

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.

Oppgave 4

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.