

**NTNU**  
**Norges teknisk-naturvitenskapelige**  
**universitet**

**Fakultet for informasjonsteknologi,**  
**matematikk og elektroteknikk**

**Institutt for datateknikk**  
**og informasjonsvitenskap**



Kontaktperson under eksamen:  
IDI, 73593440

**Kontinuasjoneksamen i TDT4186 Operativsystemer**  
**14. august 2006, 1500-1900**

Typegodkjent lommekalkulator med tomt minne tillatt  
Ingen trykte eller håndskrevne hjelpemidler tillatt

Det ønskes korte og konsise svar på hver av oppgavene.  
Det vesentlige er å kunne dokumentere forståelse,  
beherske prinsipper og se sammenhenger - ikke å kunne  
gjengi en mengde detaljer.

Der det synes å mangle noen opplysninger, må det angis  
hvilke antagelser som synes å være naturlige. Merk at  
viktige begreper er angitt på både norsk og engelsk.

**Oppgave 1: Definisjoner – 20 %**

- a) Identifiser klart hva et sentralisert system er (Centralized system)
- b) Identifiser klart hva et distribuert system er (Distributed system)
- c) Identifiser klart hva et multiprosessor system er (Multi processor system)

**Oppgave 2: Ja/Nei/Ubesvart spørsmål – 30 %**

Her følger åtte utsagn som hver for seg er riktig eller galt. Velger du å angi sannhetsverdien til et gitt utsagn, gir et riktig svar +1 poeng, mens et galt svar gir -1 poeng. Hvis du derimot velger å ikke kommentere et gitt utsagn, gir dette 0 poeng for det spørsmålet. Hvert svar kan følges av en kort og konsis begrunnelse.

- a) Prosesser (Processes)

Tråder (threads) brukes i stedet for prosesser (processes) i alle de nyeste operativsystemene !?

- b) Prosessorer (Processors)

Virkelig parallellitet (true concurrency) i en datamaskin krever flere sentrale prosesseringsenheter (CPUs – central processing units) !?

- c) Synkronisering (Synchronization)

Synkroniseringsmekanismene semaforer (semaphores) og monitorer (monitors) kan løse de samme problemene !?

- d) Vranglås (Deadlock)

Bankieralgoritmen (banker's algorithm) brukes for å umuliggjøre vranglås (deadlock prevention) !?

- e) Tidsstyring (Scheduling)

'Korteste totaltid først' algoritmen (SPN – shortest process next) er den beste algoritmen for tildeling av en prosessor (process scheduling) for alle sanntidssystemer (realtime systems) !?

- f) Lager (Memory)

'Optimal' algoritmen (OPT – optimal) for sideombytting (page replacement) er implementert i flere av de nyeste operativsystemene !?

## g) Innlesing/Utskrift (Input/Output)

'Toveis heis' algoritmen (SCAN – scanning) gir bedre midlere tjenestetid (average service time) enn 'Enveis heis' algoritmen (CSCAN – circular scanning) for tidsstyring av disk (disk scheduling) !?

## h) Implementering (Implementation)

Flere av de nyeste operativsystemene er både mikrokjerne-basert (micro kernel based) og objektorientert (object oriented) !?

**Oppgave 3: Sentrale tema – 50 %**

## a) Konstruksjon

Du er ansatt som seniorutvikler i UniqueSW. Din sjef gir deg en dag i oppdrag å konstruere et nytt operativsystem for en ny datamaskin fra UniqueHW.

Angi hvordan du vil gå fram:

- Hvilke kritiske veivalg vil du stå overfor
- Hvilke alternative løsninger vil du ha å velge mellom for hvert av disse veivalgene
- Hvilke viktige opplysninger trenger du å framskaffe for å foreta de tilhørende veivalgene

## b) Synkronisering

Et forsøk på å implementere en monitor (monitor) med semaforer (semaphores) er som følger:

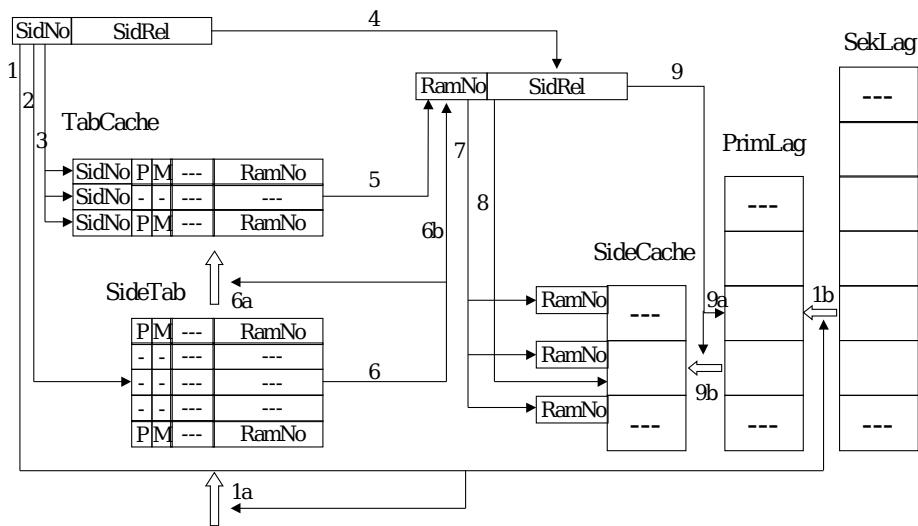
|  |  |
|--|--|
| <p><b>Variable:</b></p> <p>M: Semaphore := 1;<br/>           N: Semaphore := 0;<br/>           NC: Integer := 0;<br/>           C: Array [1..V] of Semaphore := V*0;<br/>           CC: Array [1..V] of Integer := V*0;</p> <p><b>EnterMonitorProcedure:</b></p> <p>Wait (M)</p> | <p><b>ConditionWait (C[I]):</b></p> <p>CC[I] := CC[I] - 1;<br/> <b>If</b> NC &gt; 0<br/>               <b>Then</b> Signal (M)<br/>               <b>Else</b> Signal (N);<br/>           Wait (C[I]);<br/>           CC[I] := CC[I] + 1;</p> <p><b>ConditionSignal (C[I]):</b></p> <p><b>If</b> CC[I] &lt; 0 <b>Then</b><br/>               <b>Begin</b><br/>                   NC := NC + 1;</p> |
|--|--|

|  |  |
|--|--|
| <p><b>ExitMonitorProcedure:</b></p> <p><b>If</b> NC &gt; 0<br/>             <b>Then</b> Signal (M)<br/>             <b>Else</b> Signal (N)</p> | <p>Signal (C[I]);<br/>         Wait (N);<br/>         NC := NC - 1</p> <p><b>End</b></p> |
|--|--|

Begrunn hvorfor dette er en korrekt implementasjon – eller hvorfor dette ikke er en korrekt implementasjon.

c) Lager

En totaloversikt over et sidebasert (page based) virtuelt lager (virtual memory) inkludert caching på ulike nivåer er som følger:



- SekLag* = Sekundærlager (Secondary Memory)
- PrimLag* = Primærlager (Primary Memory)
- SideCache*: PageCache
- SideTab*: PageArray
- TabCache*: ArrayCache
- SidNo* = SideNummer (Page Number)
- SidRel* = SideRelativt (Page Relative)
- RamNo* = RammeNummer (Frame Number)
- P*: Present bit
- M*: Modified bit

Forklar alle nummererte tynne / tykke piler med kopling til ulike lagerreferansehendelser (memory referance events).