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

**Fakultet for informasjonsteknologi, matematikk og elektroteknikk**

**Institutt for datateknikk og informasjonsvitenskap**

**Løsning på kontinuasjonseksamen i TDT4190 Distribuerte systemer**
**19. august 2006, 0900-1300**

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 distribuert system er (Distributed system)

SVAR: Et distribuert system består av et sett med noder som mangler felles klokke og felles lager.

b) Identifiser klart hva et distribuert multimediasystem er (Distributed multimedia system)

SVAR: Et distribuert multimediasystem er et distribuert system med interaktive MM-applikasjoner som krever svært korte tur-retur tider.

c) Identifiser klart hva distribuert delt lager er (Distributed shared memory)

SVAR: Distribuert delt lager tilsier en simulering av felles lager i en omgivelse uten felles lager.

d) Identifiser klart hva distribuert sikkerhet er (Distributed security)

SVAR: Distribuert sikkerhet berører forespørsler og resultater som kan gå via meldinger over ett / flere nett hvor de kan lett kan utsettes for ulike former for manipulasjon (aksess og endring i ulike former), og hvor kontroll av ulike aspekter blir vanskeligere på tvers av noder (identitet og rettigheter generelt sett)

## 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) Systemer (Systems)

Et multiprosessor system er et distribuert system !?

SVAR: Galt

b) Modeller og standarder (Models and standards)

CORBA er et kommersielt produkt som kan forbedre de fleste distribuerte system !?

SVAR: Galt

c) Kommunikasjon og synkronisering (Communication and synchronization)

Logiske klokker og vektorklokker kan løse de samme problemene !?

SVAR: Galt

d) Distribuerte databasesystemer (Distributed database systems)

Vranglås er mye vanskeligere å håndtere i distribuerte enn i sentraliserte system !?

SVAR: Riktig

e) Distribuerte filsystemer (Distributed file systems)

Datacaching bør utnyttes, men datareplisering bør begrenses !?

SVAR: Riktig

f) Distribuert pålitelighet (Distributed reliability)

Ulike krav til atomiskhet og ulike krav til ordnethet kan kombineres fritt !?

SVAR: Riktig

g) Distribuert navnetjeneste (Distributed name service)

Iterativ navigering mot navn utnytter datareplisering og ikke datacaching !?

SVAR: Galt

h) Implementering (Implementation)

Full konsistens og full transparens kan oppnås samtidig i distribuerte system !?

SVAR: Galt

## 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 distribuert system med nye datamaskiner 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

SVAR:

- Type: Mengde avstand?; Mengde autonomi?; Mengde heterogenitet?
- Organisering: Hvor levende?; Hvor åpent?; Hvor transparent?; Hvor konsistent?
- Form: Grad av pålitelighet?; Grad av sikkerhet?

- Nødvendig info uansett: Hvem, hva, hvordan, hvorfor
  dvs.fra funksjonalitet, via ytelse til brukergrensesnitt (og HW-kunnskap)

b) Kommunikasjon og synkronisering

En punktvis oversikt over ringalgoritmen (ring algorithm) for primas utvelgelse (process election) i et distribuert system er som følger:

**Start Valg-rutine**:
1a) Send Valg-melding m/ Egen identifikator

**Får Valg-melding**:
2a) Med høyere identifikator og Egen ikke inkludert / Egen inkludert:
Send videre m/ Gitt identifikator
2b) Med lavere identifikator og Egen ikke inkludert:
Send videre m/ Egen identifikator
2c) Med lavere identifikator og Egen inkludert:
Stopp Valg-melding
2d) Med samme identifikator og Egen inkludert:
Send Sjef-melding m/ Egen identifikator

**Får Sjef-melding**:
3a) Med høyere identifikator: Send videre m/ Gitt identifikator
3b) Med samme identifikator: Avslutt rutine

Konkrete spørsmål:

- Forklar hvordan denne algoritmen totalt sett virker

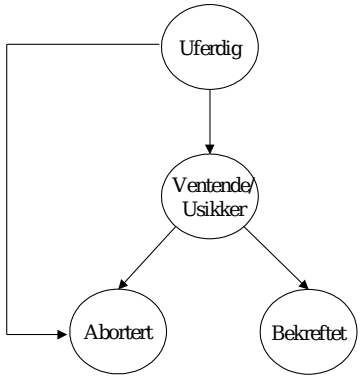
- Angi hva som oppnås med hvert av de nummererte punktene

SVAR:

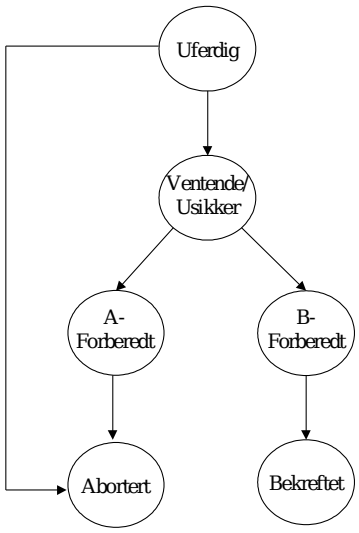
- Overordnet: Sikre at éns (med høyeste prioritet) melding vandrer helt rundt
- Detaljert: Sikre andres (med lavere prioritet) meldinger stoppes raskest mulig

c) Distribuerte databasesystemer og distribuert pålitelighet

Et forsøk på å beskrive tilstander og tilhørende tilstandsoverganger i henholdsvis 2-fase-bekreft (2PC – 2Phase Commit) protokollen og 3-fase-bekreft (3PC – 3Phase Commit) protokollen er som følger:




2PC - Koordinator / Deltagere




3PC - Koordinator / Deltagere

Konkrete spørsmål:

- Begrunn hvorfor dette er korrekt beskrivelse av 2PC / 3PC – eller hvorfor dette ikke er en korrekt beskrivelse av 2PC / 3PC - for en situasjon der verken terminering (i forbindelse med at en node går ned) eller gjenoppretting (i forbindelse med at en node kommer opp igjen) er tatt med

- Beskriv eventuelle endringer (med hensyn til tilstander og tilhørende tilstandsoverganger) som må til for 2PC / 3PC når henholdsvis nodefeil og kommunikasjonsfeil tas med (med henblikk på både terminering og gjenoppretting)

SVAR:

- Uten terminering / gjenoppretting:

  2PC: OK
  3PC: Ekskluder AF-tilstand

- Med terminering / gjenoppretting:

  Kun nodefeil;
  2PC: OK

3PC: Ekskluder AF-tilstand + Innkluder 2 ekstra overganger
(VU->B, BF->A)

Node- + Komm.feil;
2PC: OK
3PC: Innkluder 7 ekstra overganger
(VU->B, BF->A; VU->AF, BF->AF, AF->BF, AF->B, AF->A)