



EKSAMENSOPPGAVE I SIE5025-PÅLITELIGE SYSTEMER

Faglig kontakt under eksamen: Bjarne E. Helvik
Telefon.: 92667

Eksamensdato: 25. mai 2002
Eksamenstid: 4 timer
Vekttall: 2,5 Vt
Tillatte hjelpemidler: D

Språkform:

Antall sider bokmål: 4

Antall sider nynorsk:

Antall sider engelsk:

Antall sider vedlegg:

Sensurdato¹: 17 juni 2002

1. Merk! Studentene må primært gjøre seg kjent med sensur ved å oppsøke sensuroppslagene. Evt. telefoner om sensur må rettes til sensurtelefonene. Eksamenskontoret vil ikke kunne svare på slike telefoner.

Oppgave 1

[Oppgaven tillegges 50% vekt]

- Beskriv kort (ca. fem linjer pr. metode) tre ulike metoder for å prediktere antall logiske feil i programvarekode. Prediksjonene skal være basert på kun selve koden.
- Beskriv kort en modell av den prosessen som finner sted når en logisk feil i kontinuerlig eksekverende programvare fører til en feilytring.

Betrakt en programvare som idriftsettes ved tiden $t = 0$. Denne inneholder logiske feil. Raten logiske feil aktiveres med λ . Gitt at en logisk feil aktiveres, vil den gi en øyeblikkelig feilytring med sannsynlighet p . Gitt at det introduseres tilstandsfeil i programvaren, vil dette resultere i en feilytring etter en tid T_α . Varigheten av denne tiden er gitt av sannsynlighetstetthetsfunksjonen $f_{T_\alpha}(\tau) = \alpha e^{-\alpha\tau}$. Alle tider til aktivering av logiske feil og tider fra tilstandsfeil oppstår til de fører til feilytringer er uavhengige.

- Hva kalles tidene som er gitt av sannsynlighetstetthetsfunksjonene $\lambda e^{-\lambda\tau}$ og $p \cdot \delta(\tau) + (1 - p)\alpha e^{-\alpha\tau}$? (Både engelske og fullstendige og beskrivende norske betingelser godtas.)¹
- Tegn opp et tilstandsdiagram med sikte på å bestemme funksjonssannsynligheten til programvaren beskrevet over. Etabler et fullstendig ligningsett for å finne denne funksjonssannsynligheten. (Ligningsettet kreves ikke løst.)

Løst gir dette ligningsettet:

$$R(\tau) = \frac{1}{\alpha - \lambda} ((\alpha - \lambda p)e^{-\lambda\tau} - \lambda(1 - p)e^{-\alpha\tau})$$

- Finne et uttrykk for feilraten til programvaren, $\lambda(\tau)$.

Anta at når programvaren feiler blir den umiddelbart restartet. Restartprosessen tar en neglisjerbar tid, dvs. at programvaren kan antas å være "som ny" umiddelbart etter en feil. Etter en restart er programvaren og omgivelsene (statistisk sett) de samme som da programvaren ble startet første gang.

- Finne et uttrykk for feilintensiteten til programvaren når den er stasjonær, dvs. at tiden siden programvaren ble startet første gang er uendelig lang.
- Vis at feilraten til programvaren når tiden går mot uendelig, $\lim_{\tau \rightarrow \infty} \lambda(\tau)$, er ulik feilintensiteten i deloppgave f) når $\alpha \neq \lambda$. Gi en meget kort fysikalsk fortolkning dette og av $\lim_{\tau \rightarrow \infty} \lambda(\tau)$ for de to tilfellene $\alpha > \lambda$ og $\alpha < \lambda$.

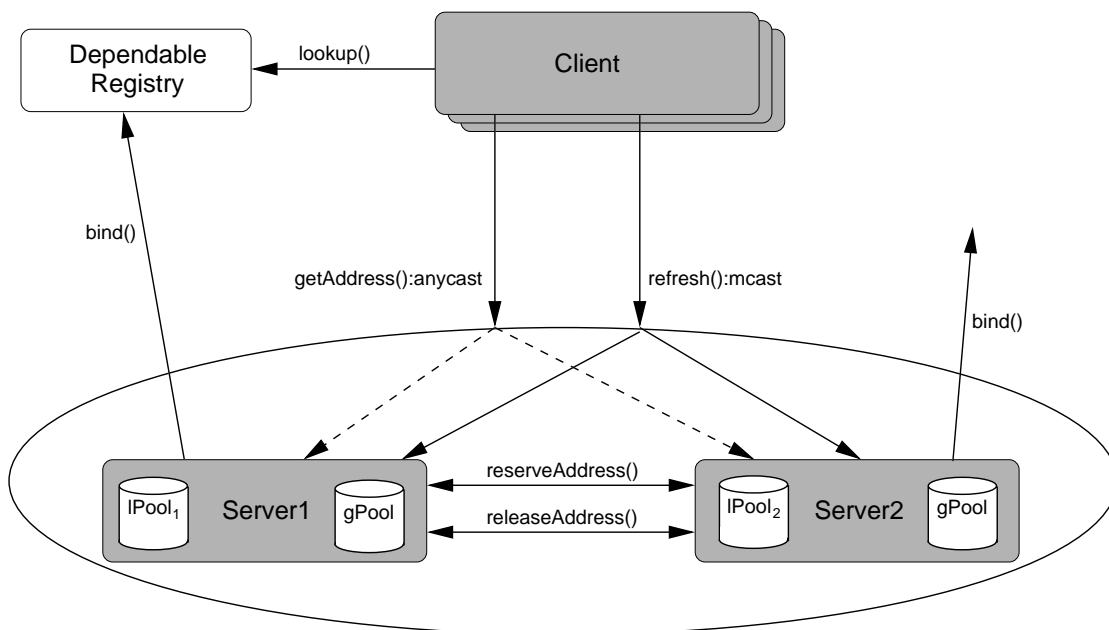
Tips: I forenklingen av uttrykket, benytt at $Ae^{-\lambda\tau} + Be^{-\alpha\tau} = e^{-\lambda\tau}(A + Be^{(\lambda-\alpha)\tau})$.

1. $\delta(\tau)$ angir Dirac funksjonen: $\delta(0) = \infty$, $\delta(x) = 0$ når $(x \neq 0)$, $\int_{-\infty}^{\infty} \delta(\tau) d\tau = 1$

Oppgave 2

[Oppgaven tillegges 50% vekt]

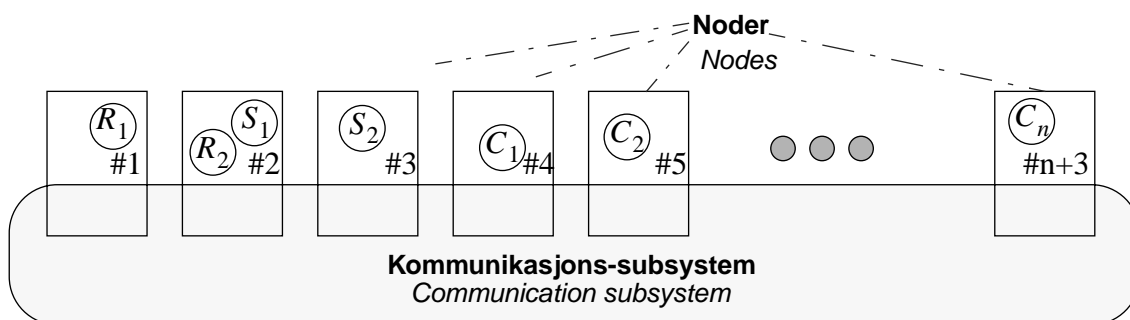
I denne oppgaven betraktes et klient-tjenersystem tilsvarende laboratorieoppgaven i emnet SIE5025 våren 2002. De to tjenerne er lastdelte og låner ut (“leaser”) nettadresser til klienter. Hvert tjenerreplika har ansvaret for en del av adresseområdet. Dersom en tjener feiler vil den andre overta ansvaret for dennes adresseområde. Lånte (“leased”) adresser må gjenoppfriskes (“refreshes”) etter en tid. Tjenerne er organisert som en gruppe og systemet er realisert i Jgroup, se illustrasjon i Figur 2.1



Figur 2.1

- En av målsettingene med gruppekommunikasjon er at bruk av replikerte tjenere (“servers”) fra klienter (“clients”) skal være så likt å bruke en ureplikert tjener som mulig. Ta utgangspunkt i en “vanlig” ureplikert tjener som aksesseres vha. Java RMI. La tjeneren bli replikert og tilgjengelig som en gruppe i Jgroup. Hva er de to vesentligste endringene som må gjøres i klienten for at den skal samspille med gruppen av tjenerreplika?
- Hva er “anycast” og hva er “multicast”? Forklar hvorfor de benyttes som de gjør til å få adresser (getAddress) og til å gjenoppfriske adresser (“refresh”) i Figur 2.1.

Klient-tjenersystemet vist Figur 2.1 er implementert i et distribuert system som illustrert på Figur 2.2. Systemet består av en rekke noder knyttet sammen ved hjelp av et feilfritt kommunikasjons-subsystem. I systemet er det to replika av tjenerprosessen S_i , $i = 1, 2$, to replika av “dependable registry” R_i , $i = 1, 2$, samt en rekke klienter C_i . Disse er allokkert til noder som vist i Figur 2.2, dvs. ett replika av “dependable registry” på node #1, ett replika av både “dependable registry” og tjenerprosessen på node #2, og ett replika av tjenerprosessen på node #2. Et replika kan ta hele lasten.



Figur 2.2

- c) Hvilken feilsemantikk bør vi ha for nodene #1, #2 og #3 (for maskinvare, operativsystem, osv.) for at tjenerdelsystemet skal være feiltolerant?

Hver av nodene #1, #2 og #3 feiler med en konstant feilrate λ . Kun en node kan repareres ad gangen. Reparasjonsraten er μ . Ved flere samtidige feil, repareres den feilte noden først som vil bringe systemet tilbake til en arbeidende tilstand eller, dersom begge/alle alternativene gjør det, den som gir størst robusthet mot nye feil.

- d) Tegn et tilstandsdiagram (Markov modell) som kan benyttes for å bestemme tilgjengeligheten av tjenerdelsystemet bestående av nodene #1, #2 og #3.

Anta i det etterfølgende at nodene #1, #2 og #3 repareres uavhengig av hverandre med reparasjonsrate μ .

- e) Tegn et blokkskjema og bestem utilgjengeligheten for tjenerdelsystemet bestående av nodene #1, #2 og #3.
- f) Finn midlere tid mellom feil og midlere nedetid for tjenerdelsystemet. Begrunnede tilnærmede løsninger basert på at $\lambda \ll \mu$ er akseptable.
- g) Node #1 har kapasitet til å kjøre både R_1 og S_2 . Systemet rekonfigureres slik at tjenerdelsystemet blir bestående av nodene #1 og #2. Blir utilgjengeligheten til tjenerdelsystemet større eller mindre av denne rekonfigureringen? Finn et uttrykk for den relative endringen. I hvilket område vil denne ligge hva blir den tilnærmet når $\lambda \ll \mu$? Gi en kort fysikalsk fortolkning.