2.5 Fault tolerance

Thursday, April 2, 2020 4:47 PM

Feiltoleranse forbindes med systemets egenskap til å fortsette normal operasjon til tross for hardware- eller software-feil.

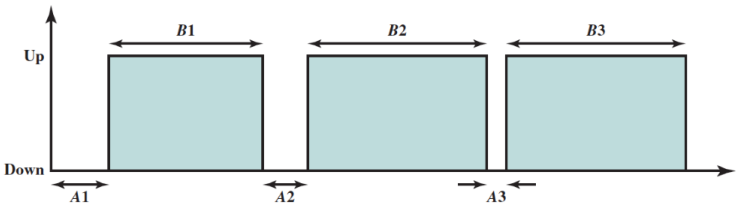
Grunleggende konsepter

Det er tre grunnleggende mål for kvaliteten av feiltoleranse; *reliability, mean time to failure (MTTF),* og *availability*. Disse konseptene gjelder egentlig hardwarefeil, men de gjelder generelt både hardware- og softwarefeil.

- **Reliability *R(t)* (pålitelighet):** Er definert som sannsynligheten for den korrekte operasjonen opptil tidspunkt *t* gitt at systemet opererte korrekt ved tidspunkt *t* = 0. For computer systemer og operativsystemer betyr *korrekt operasjon* den korrekte utføringen (execution) av et sett av programmer, og beskyttelsen av data fra unønsket modifisering.
- **Mean time to failure (MTTF):** Er definert som

  $$MTTF = \int_0^\infty R(t)\,dt$$

  Mean time to repair (MTTR) er den gjennomsnittlige tiden det tar å fikse eller erstatte et feilelement. Figuren under viser sammenhengen mellom MTTF og MTTR.



$$MTTF = \frac{B1 + B2 + B3}{3} \qquad MTTR = \frac{A1 + A2 + A3}{3}$$

Figure 2.13 System Operational States

- **Availability (tilgjengelighet):** Tilgjengeligheten av et system eller en tjeneste er definerst som brøkdelen av tid systemet er tilgjengelig for tjenestegjøring av brukers forespørsler. Tilgjengeligheten er sannsynligheten for at en entitet opererer korrekt under gitte omstendigheter ved et gitt tidspunkt. Tiden systemet ikke er tilgjengelig kalles *downtime*; tiden den er tilgjengelig kalles *uptime*. Tilgjengeligheten *A* av et system kan uttrykkes på følgende måte:

$$A = \frac{MTTF}{MTTF + MTTR}$$

Feil

En feil er definert som en feilaktig hardware eller software tilstand med opphav i enten en komponentfeil, operatorfeil, fysisk forstyrrelse fra miljø, designfeil, programfeil eller en datastrukturfeil. Vi kan gruppere de ulike feilene inn i følgende kategorier:

- **Permanent:** En feil som alltid vil være tilstede etter den har skjedd. Feilen er der inntil den feilaktige komponenten er erstattet eller reparert.
- **Midlertidig:** En feil som ikke alltid er tilstede for alle operasjonsomdstendigheter. Midlertidige feil kan igjen kategorises:
  - **Transient (flyktig):** En feil som kun skjer én gang.
  - **Intermittent (periodisk):** En feil som kan skje ved flere uforutsigbare tilfeller.

Generelt er feiltoleranse bygget inn i systemet ved redundans-metoder:

- **Spatial (fysisk) redundans:** Fysisk redundans involverer bruk av flere komponenter som enter gjør samme funksjon flere ganger, eller er konfigurert slik at en komponent alltid er tilgjengelig som en backup i tilfelle en annen komponent streiker.
- **Midlertidig redundans:** Midlertidig redundans involverer repitering av en funksjon eller operasjon når en feil blir oppdaget. Denne tilnærmingen er effektiv ved midlertidige feil, men ikke for permanente feil.
- **Informasjonsredundans:** Informasjonsredundans tilbyr feiltoleranse ved å replikere eller kode data på en måte som gjør at bit-errorer kan både bli oppdaget og korrigert.

Operativsystem mekanismer

Det er flere teknikker som kan inkorporeres i OS-softwaren for å støtte feiltoleranse. Her er en liste med eksempler:

- **Prosess-isolasjon:** Som nevnt tidligere er prosesser generelt isolert fra hverandre i forbindelse med hovedminnet, filaksess, og flyt av execution. Strukturen tilbudt av OS-et for å håndtere prosesser tilbyr til en viss grad beskyttelse fra andre prosesser fra en prosess som eventuelt produserer feil.
- **Samtidighetskontroller:** Kapittel 5 og 6 ser nærmere på problemer som oppstår når prosesser samarbeider og kommuniserer. Disse kapitlene ser på teknikker for å håndtere blant annet vranglåser.
- **Virtuelle maskiner:** Se kapittel 14 (ikke pensum) for nærmere informasjon. Virtuelle maskiner tilbyr en større grad av applikasjon-isolasjon og dermed feil-isolasjon. Virtuelle maskiner kan også tilby redundans-metoder, hvor en virtuell maskin fungerer som en backup for en annen.
- **Sjekkpunkter og rollbacks (tilbakerulling):** Et sjekkpunkt er en kopi av en applikasjons tilstand lagret i et lagringsmedium som er immun mot feilene under betraktning. En rollback restarter en execution fra et tidligere lagret sjekkpunkt. Når en feil oppstår, så rulles applikasjonstilstanden tilbake til det tidligere sjekkpunktet og restarter fra der. Denne teknikken kan brukes til å gjenopprette fra flyktige, men også permanente hardwarefeil, og noen typer softwarefeil. Database og transaksjons prosesseringssystemer har typisk egenskaper for dette innebygget.