SIF8041 OPERATIVSYSTEMER OG DATABASER

våren 2001

ØVING 5 : Transaksjoner

LØSNINGSFORSLAG

a) Hva er definisjonen på at en gruppe transaksjoner blir korrekt utført ?

Utførelsen skal være ekvivalent med en seriell utførelse.

b) Vil følgende sekvens av operasjoner gi korrekt utførelse?

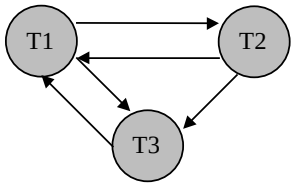
r1(x), r2(y), r1(z), w2(y), r3(x), r3(y), w1(z), r1(y), w3(x), w1(y), r2(z), c1, c2, c3.

ri(a) betyr at transaksjon i leser ressurs a. wi(a) betyr at transaksjon i skriver ressurs a. ci betyr commit for transaksjon i.

r1(x), r2(y), r1(z), w2(y), r3(x), r3(y), w1(z), r1(y), w3(x), w1(y), r2(z), c1, c2, c3.
Konflikter om x: T1 < T3

r1(x), r2(y), r1(z), w2(y), r3(x), r3(y), w1(z), r1(y), w3(x), w1(y), r2(z), c1, c2, c3.
Konflikter i y: T2 < T3, T2 < T1, T3 < T1

r1(x), r2(y), r1(z), w2(y), r3(x), r3(y), w1(z), r1(y), w3(x), w1(y), r2(z), c1, c2, c3.
Konflikter i z: T1 < T2



De forskjellige betingelser er illustrert i figuren over og det er ikke mulig å tilfredstille alle betingelser gjennom en seriell utførelse. Utførelsen er altså ikke korrekt.

c) *Forklar hvorledes metoden: "no undo - redo" kan brukes til å sikre konsistent utførelse. Forklar spesielt hvordan blokkbuffer og loggen blir brukt i sammenhengen.*

For å kunne unngå UNDO må vi vente med å føre endringer inn i databasen til etter commit. Endringene kan ligge i buffer og hvis buffer blir for lite må endringene skrives til et hjelpelager. Før commit må alle endringer være skrevet til REDO-loggen. Før commit er altså endrede bufre låst mht. skriving til databasen. Ved commit fjernes denne låsen. Ved abort må endrede blokker invalideres slik at de ikke skrives til databasen. Metoden er svært effektiv så lenge en har korte transaksjoner i forhold til hva det er plass til av bufre i arbeidslager. Dermed slipper vi skriving til hjelpelager.