

6.2 Deadlock prevention

Saturday, April 18, 2020 7:19 PM

Umuliggjøring omhandler strategien for å bygge systemer der muligheten for at det oppstår vranglåser ikke eksisterer. Dette er åpenbart vanskelig (kanskje umulig i kompliserte systemer) å gjennomføre, men det finnes hovedsaklig to metoder: indirekte metoder (dette gjelder de tre første betingelsene på forrige side) og direkte metoder (betingelse nr. 4). Vi ser nærmere på disse fire teknikkene:

Indirekte metoder

Gjensidig utelukkelse

Denne betingelsen kan ikke elimineres. Hvis aksess til en ressurs krever gjensidig utelukkelse så må det være støttet av OS-et. Noen ressurser som eksempelvis filer kan tillate flere aksesser for lesing, men kun én eksklusiv aksess for skriving. Selv i dette tilfellet kan vranglåser oppstå hvis flere prosesser påkrever skrive-tillatelse.

Hold og vent

Denne betingelsen kan unngås ved at prosesser må etterspørre alle sine nødvendige ressurser på én gang og blokkeres inntil alle forespørselene kan imøtekommes. Denne tilnærmingen er ineffektiv av to grunner. For det første må en prosess kanskje vente veldig lenge selv om den kunne ha kjørt med bare noen av ressursene den etterspurte. For det andre kan det hende at noen av ressursene tildelt en prosess kan ha lite brukstid, og dermed forblir låst lenge selv om de er inaktive. Dette er problem hvis andre prosesser etterspør disse gitte ressursene. I tillegg kan det hende at en prosess ikke vet hvilke ressurser de krever i forkant. Det er også en praktisk problem ved bruk av modulær programmering eller for multitrådede strukturer i en applikasjon.

Ingen avbrudd

Denne betingelsen kan forebygges på flere måter. Hvis en prosess som holder en gitt ressurs blir avvist ytterligere forespørsler så må prosessen frasi seg de opprinnelige ressursene de har brukt, og hvis nødvendig etterspørre dem på nytt med den ekstra ressursen. På samme måte kan en prosess som etterspør ressurser fra en nåværende kjørende prosess bli bedt om å frasi seg sine ressurser av OS-et. Denne siste varianten vil forebygge en vranglås kun hvis prosessene hadde ulik prioritet. Denne tilnærmingen er kun effektiv når tilstanden til ressursene lett kan lagres og innhentes (som ved en prosessor).

Sirkulær venting

Denne betingelsen kan forebygges ved å definere en lineær ordning av ressurstyper. Dersom en prosess har fått tildelt en ressurs av type R , så kan den kun etterspørre andre ressurser av typen som følger R sin rekkefølge. I likhet med hold-og-vent kan forebygging av sirkulær venting være ineffektivt og på unødvendigvis senke prosesser og forhindre ressursaksess.