

5.2 Principles of concurrency

Thursday, April 9, 2020 12:19 AM

I et enkelt-prosessor multiprogrammerende system er prosesser innflettet i tid til at det virker som de har samtidig execution. Selv om det ikke forekommer faktisk parallel prosessering så er det en viss del overhead som går til å bytte frem og tilbake mellom prosesser. Innfletting av execution gir store fordeler i prosesseringseffektiviteten og i programstrukturering. I et multiprosessorsystemer er det ikke bare mulig å innflette execution av flere prosesser, men også å overlape dem. Begge metodene kan dog sees på som samtidig prosessering.

Problemet for multiprogrammering og multiprosessering stammer fra at man *ikke* kan vite den relative hastigheten til utførelsen av en prosess. Den vil avhenge av andre prosessers aktiviteter, hvordan OS-et håndterer avbrudd og tidsstyringen til OS-et. Utfordringer som oppstår er:

- Når flere prosesser leser fra og skriver til samme globale variabler, som gjør at rekkefølgen av execution blir kritisk.
- Det er vanskelig for OS-et å håndtere tildeling av ressurser optimalt.
- Det er vanskelig å lokalisere en programmeringserror fordi resultatene typisk ikke er deterministiske og rekonstruerbare.

Utfordringer knyttet til samtidighet i operativsystemet

1. OS-et må kunne holde styr på de ulike prosessene. Dette gjøres ved bruk av prosesskontrollblokken.
2. OS-et må tildele og frigjøre ulike ressurser for aktive prosesser. Disse ressursene inkluderer:
 - Prosessortid (dette gjøres av planleggingsfunksjonen)
 - Minne (de fleste operativsystemene bruker en form for virtuelt minne)
 - Filer (dette håndteres av et fil-håndteringssystem)
 - I/O (dette håndteres typisk av I/O-arkitekturen i form av I/O-funksjoner, buffer og disk)
3. OS-et må beskytte dataen og fysiske ressurser hos hver prosess mot uønsket forstyrrelse fra andre prosesser. Dette inkluderer teknikker relatert til minnet, filer og I/O-enheter.
4. Funksjonaliteten av en prosess, output-en den produserer må være uavhengig av hastigheten execution utføres relativt til hastigheten av andre samtidige prosesser.

Prosessinteraksjon

Vi kan klassifisere måtene prosesser interagerer med hverandre:

Type	Beskrivelse	Potensielle kontrollproblemer
Konkurranse	Resultatet av en prosess er uavhengig av handlingene til andre prosesser. Timingen til prosessen kan bli påvirket. Proessen vet ikke om eksistensen av andre prosesser og må dermed ikke endre tilstanden på delte ressurser.	<ul style="list-style-type: none">• Gjensidig utelukkelse• Vranglås• <i>Starvation</i> = (En prosess får aldri kontroll over ressursen)
Samarbeid via deling	Resultatet av en prosess kan avhenge av informasjon som oppholdes av andre prosesser. Timingen til prosessen kan bli påvirket. Data kan aksesseres gjennom read/write, og kun write-operasjoner krever gjensidig utelukkelse.	<ul style="list-style-type: none">• Gjensidig utelukkelse• Vranglås• Starvation• Dataavhengigheter
Samarbeid via kommunikasjon	Resultatet av en prosess kan avhenge av informasjon som oppholdes av andre prosesser. Timingen til prosessen kan bli påvirket.	<ul style="list-style-type: none">• Vranglås• Starvation

Litt mer om prosesser i konkurranse

Det er tre kontrollproblemer som kan oppstå for prosesser i konkurranse. Det er viktig å understreke at disse prosessene ikke er klar over hverandres eksistens, så de tror de opererer alene – dermed spiller OS-et en viktig rolle i tildelingen av ressurser. Det første kontrollproblemet er derav sikring av gjensidig utelukkelse. En ressurs som mottar kommandoer fra prosesser kalles en *kritisk ressurs*, og den delen av programmet som bruker ressursen kalles den *kritiske seksjonen* av programmet. Det er viktig at kun ett program om gangen får lov til å være i sin kritiske seksjon. Dette delikate problemet overgår hva vi kan forvente av håndteringen til OS-et, så det kreves ekstra tilrettelegging for å sikre gjensidig utelukkelse.

De to andre kontrollproblemene er vranglås og såkalt *starvation*.

Krav for gjensidig utelukkelse

Enhver form for tilrettelegging av gjensidig utelukkelse bør imøtekomme følgende krav:

1. Gjensidig utelukkelse må håndheves: Kun en prosess om gangen er tillatt å være i sin kritiske seksjon, mellom alle prosessene som har kritiske seksjoner for de samme ressursene eller delte objektene. Kritiske regioner kan ikke overlappe.
2. En prosess som stanser i sin ikke-kritiske seksjon må gjøre det uten å forstyrre andre prosesser.
3. Det kan ikke være mulig for en prosess som krever aksess til en kritisk seksjon å utsettes forevig: ingen vranglås eller starvation.
4. Når ingen prosesser er i en kritisk seksjon, så må enhver prosess som forespør inngang til sin kritiske seksjon bli tillatt å entre uten forsinkelse.
5. Ingen antagelser om relativ proseshastighet eller antall prosessorer.
6. En prosess forblir i sin kritiske seksjon for en begrenset tidsperiode.