Oppgave 1

1. Et operativsystem, er et program som fungerer som et bindeledd mellom brukeren av en datamaskin og maskinvaren. Den skal gjøre maskinen tilgjengelig for brukeren, gjøre maskinvaren mest mulig effektiv og løse brukerens problemer på en enkel måte. Operativsystemet allokerer ressursene. Den holder konroll på alle ressursene og fordeler de rettferdig. Den styrer også programmer for å unngå feil under prosessen.
2. En prosess, er et program under utføring en arbeidsenhet i systemer. For at prosessen skal kunne gjøre jobben sin, trenger den ressurser. Nå en prosess er ferdig, frigjør den alle resurssene likt at de kan brukes på nye prosesser. Eksempel på bruk, er at den skal lage og fjerne både bruker- og systemprosesser. Det er også mulig å pause/starte der sluttet.
3. Tråder er mye det samme som en prosess, men tråer er mye lettere å opprette. De kan også forekle koden og dermed blir de mer effektive. VI kan også bruke flere tråder(flertrådet prosess). Det er også billigere å bytte kontekst i tråder i forhold til porsesser.
4. En kritisk region, er en plass det der kan oppstå kollisjoner. Men andre ord, er det et stykke programkode prøver å få tilgang til en ressurs som ikke kan deles. Hvis det er flere trådet som får tilgang til til samme ressurs, kan det oppstå problemer. Da vil ikke programmet fungere på den måten vi vil. og kan krasje.
5. En semafor, kan beskrives som et trafikklys. Den brukes til å holde kontroll på multiprosesser og bestemmer hvilke ressurser som skal brukes i hvilke tilfeller.
6. Velger den første prosessen i køen  
   Løkke  
    Sjekker den valgte prosessen.  
    Den trenger andre prosesser for å kjøre  
    Finner ut hvilke prosesser som trengs  
    Velger disse prosessene  
    Løkke  
    Denne trenger ikke andre prosesser for å kjøre  
    Starter prosessen  
    Velger neste prosess i køen  
    Løkke  
    Det er ikke flere prosesser igjen  
    Avslutt
7. Banksjefens algoritme, er en algoritme som skal unngå deadlock. Her sjekker algoritmen hvor mange ressurser den trenger (maksimal) for å kunne kjøre denne prosessen. Deretter sjekker den hvor mange ressurser som er ledig. Hvis prosessen ikke overskrider antall ledige ressurser, kan vi starte prosessen. Hvis ikke, må den vente til det er nok ressurser.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Allokert | Maks Behov | Trenger | Tilgjengelig |
| P1 | 010 | 753 | 743 | 230 |
| P2 | 302 | 322 | 020 |  |
| P3 | 302 | 902 | 600 |  |
| P4 | 112 | 433 | 321 |  |

i)

P1 = F

P2 = T - 532

P3 = F

P4 = T - 644

P1 = F

P3 = T - 946

P1 = T - 956

Path <P2,P4,P3,P1>

ii)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Allokert | Maks Behov | Trenger | Tilgjengelig |
| P1 | 010 | 753 | 743 | 020 |
| P2 | 302 | 322 | 020 |  |
| P3 | 302 | 902 | 600 |  |
| P4 | 322 | 433 | 111 |  |

P1 = F

P2 = T - 322

P3 = F

P4 = T - 644

P1 = F

P3 = T - 946

P1 = T - 956

Path <P2,P4,P3,P1>

iii)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Allokert | Maks Behov | Trenger | Tilgjengelig |
| P1 | 110 | 753 | 643 | 120 |
| P2 | 302 | 322 | 020 |  |
| P3 | 302 | 902 | 600 |  |
| P4 | 112 | 433 | 321 |  |

P1= F

P2= T - 422

P3=F

P4=T - 534

P1=F

P3=F

Stopper

Fungerer ikke