Oppgaver relatert til “The programming interface/Concurrency and threads”

## Oppgave 1. Generelt om opsys og om prosesser (vekt: 25%)

1. Vi sier at operativsystemet har to viktige roller. Den ene er å være ressursadministrator. Den andre er å være et abstraksjonslag slik at programvare kan kjøre på maskinen uten å måtte kjenne til maskinvaredetaljer. Forklar hva dette abstraksjonslaget er for noe.
2. I en enkel modell av prosessene kan en prosess være i 3 tilstander. Hvilke tilstander er det?

-Klar: Prosessen er klar til kjøring og trenger bare tilgang til CPU-en, den står i det vi kaller en CPU-kø. Det er forutsatt at prosessen har tilgang til alle andre ressursene den trenger bortsett fra CPU-en.

-Kjørende: Dette er tilstanden hvor prosessen er kjørende på CPU-en. Prosessen blir i denne tilstanden til 1 av 3 ting skjer:

1. Prosessen er ferdig og avsluttes.

2. Operativsystemet mener prosessen har kjørt for lenge, og setter den tilbake i cpu-køen og slipper en ny prosess til.

3. Prosessen trenger tilgang til en ressurs og blokkeres fram til den har fått tilgang, dette fører til at den ikke lenger er i kjørende tilstand.

-Blokkert

Her venter prosessen på at ressursen den trenger skal bli ledig, og når den får tilgang til ressursen stiller den seg tilbake i klar tilstand (kan ikke gå direkte til kjørende).

1. Det er 4 mulige overganger mellom disse tilstandene nevnt i oppgaven ovenfor. Hvilke er det og hva er det som fører fra en tilstand til en annen?

-En prosess kan gå fra klar til kjørende når den har tilgang til ressursene den trenger og bare mangler cpu-en.

-Om prosessen kjører for lenge kan den bli satt tilbake i klar tilstand slik at flere prosesser kan kjøre på cpu-en

-Om prosessen kjører og mangler en ressurs vil prosessen gå over i blokkert tilstand.

-Når prosessen har fått tilgang til ressursen går den fra blokkert tilstand til klar tilstand.

1. En prosess kan splitte kjøringen opp i flere tråder. Forklar hvorfor det å operere med flere tråder innenfor en prosess er hensiktsmessig.

-Grunnen til at dette er hensiktsmessig er fordi trådene ikke knytter til seg noen ressurser, og det er derfor mindre krevende og enklere å opprette nye tråder, noe som gjør at responstiden blir kortere. Med mange aktive tråder som betjener hver sin forespørsel vil CPU-tida bli mye mer effektivt utnyttet. De gir også mulighet for å kjøre flere aktiviteter samtidig, der alle aktivitetene har tilgang til samme adresseområdet.

## Prosesser og tråder

## En fordel med å bruke prosesser fremfor tråder i et program er:

## Velg ett alternativ

* at det blir raskere for CPU-en å skifte kontekst.
* at det er raskere å opprette en ny prosess enn en ny tråd.
* at det blir brukt mindre minne.
* hvis en prosess krasjer, så krasjer ikke hele programmet. (riktig)

## Årsaken til at et program oppretter flere tråder i en prosess er:

## Velg ett alternativ

* å dele minne mellom prosesser.
* å benytte seg av avbrudd.
* å vise når prosessen er ferdig (exit() systemkall).
* å gjøre flere ting samtidig, samt å dele ressurser innen prosessen. (riktig)

## Prosesstabellen

## Prosesstabellen inneholder blant annet følgende informasjon:

## Velg ett alternativ

* Instruksjonspeker, prosess ID, stakkpeker, tråd ID
* CPU-registrer, instruksjonspeker, stakkpeker, tråd ID
* CPU-registrer, instruksjonspeker, prosess ID, foreldre-prosess ID
* Instruksjonspeker, prosess ID, CPU-tid brukt, barneprosess ID

## Oppgave 4. Prosesser og tråder (vekt: 25%)

1. Vi snakker ofte om både prosesser og tråder. Forklar hva en prosess er og hva en tråd er og hvilken sammenheng det er mellom prosesser og tråder.

En prosess er kjørende program. Ved kjøring av et program oppretter operativsystemet en prosess i minnet med eget data- og instuksjonsområdet. En tråd er det som kjører i prosessene, og alle prosessene har minst en tråd. Men de kan inneholde flere tråder, der alle trådene har sin egen stakk og dataområde, men med felles instruksjoner. Alle trådene deler på prosessens dataområde noe som gjør at de kan utveksle informasjon og samarbeide. Hver prosess har sin egen PCB som inneholder PID (prosess ID), status om kjøring osv.

1. Hvordan er det mulig å kjøre flere tråder på en CPU enn det er tilgjengelige kjerner i CPU-en? Forklar.

-Pga prosessadministratoren i operativsystemet har en schduler. Denne gir hver prosess et gitt tidskvant og setter prosessen tilbake i CPU-køen etter denne tiden er oppbrukt (kvasiparallell). Dette lar CPU-en betjene mange tråder i rask rekkefølge noe som gjør at vi får simuleret simultan kjøring aka multitasking.

1. Tjenerprogrammer, f.eks på en filtjener eller på en webtjener, er som oftest såkalt ”multithreaded”, dvs kjører mange tråder. Hvorfor er det hensiktsmessig å kjøre mange tråder på f.eks en filtjener sammenlignet med å kjøre kun en prosess? Forklar.

-Om vi hadde brukt prosesser måtte vi ha ventet på at hver prosess skulle gjøre seg ferdig før neste kunne ha begynt, mens om vi bruker tråder kan vi bruker flere samtidig noe som gjør at responstiden blir kortere, og cpu-tiden blir brukt mer effektivt. Det vil derfor ta mye lenger tid om vi hadde brukt en prosess i motsetning til bruk av tråder.

1. Prosesser får tilgang til i/o-enheter via operativsystemkjernen. På hvilken måte? Forklar.

-Prosesser går i kjernemodus og spør kjernen om de får bruke i/o-utstyret (systemkall), det er dermed opp til kjernen å administrere køen. Kjernen sjekker om prosessen som etterspør i/o-utstyret har rettighetene til å bruke den og setter deretter prosessen i kø.

1. Hvorfor kan ikke prosessene gå direkte mot i/o-enhetene og gjøre sin egen i/o uavhengig av operativsystemkjernen? Forklar.

-Operativsystemets oppgave er å administrere i/o-utstyret, uten operativsystemet ville det ikke vært noe form for system på hvilken prosess som fikk bruke i/o-utstyret og hvor lengde den fikk ha den i bruk. Dette er en viktig del for at prosessene skal gå rundt siden de vil ligge i blokkert tilstand helt til de har fått tilgang til ressursene de trenger. I tillegg til å administrere køen tilbyr kjernen gode og sikre metoder for å bruke utstyret.