IN1020, Gruppe 5

Uke 5

Velkommen :)

Agenda

- Operativsystemer
- Ukesoppgaver

Eksempler på operativsystemer

+++ mer

Operativsystemer er overalt



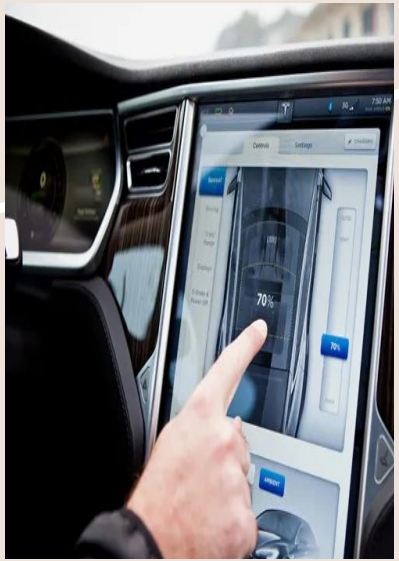
Datamaskiner



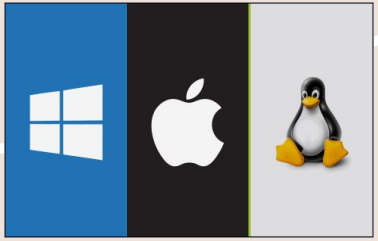
Telefoner



Gaming



Biler





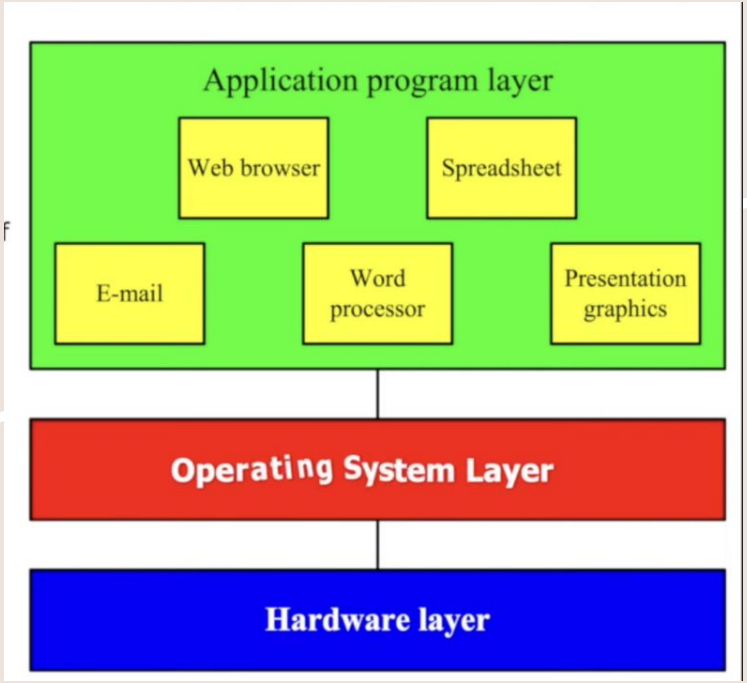


En bil har mange sensorer (som temperatur, regn, lys, varm, kameraer osv…) og har flere andre inputkilder som ratt, pedaler vifter

->Operativsystemet er mellomleddet

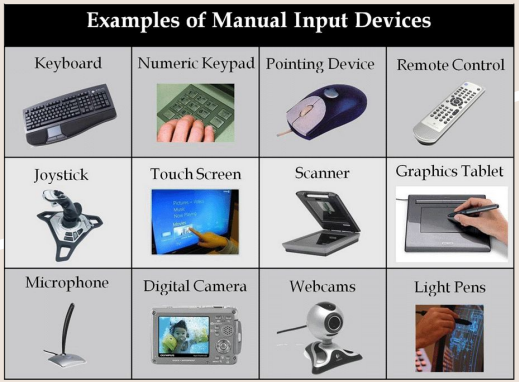


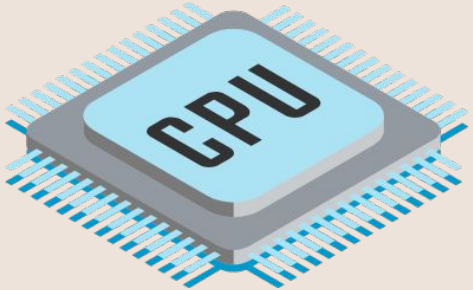
Hvilket utfordringer kan man ha når flere programmer kjøre samtidig uten OS?



Hvilket utfordringer kan man ha når flere programmer kjøre samtidig uten OS?

- Man kan ha forskjellige programmer som utnytter maskinen forskjellig. Man er ofte ikke alltid alene på en maskin.
- Noen av utfordringene man ser er hvordan man skal kunne håndtere alt dette samtidig?
- Hvordan skal jeg sortere mellom forskjellige typer oppgaver som alle trenger seg på samtidig. Og beskytte disse her fra hverandre på en effektiv og sikker måte?
- Hvordan kan man forsikre at jeg får like stor tilgang til de ressursene som er der som en annen bruker?

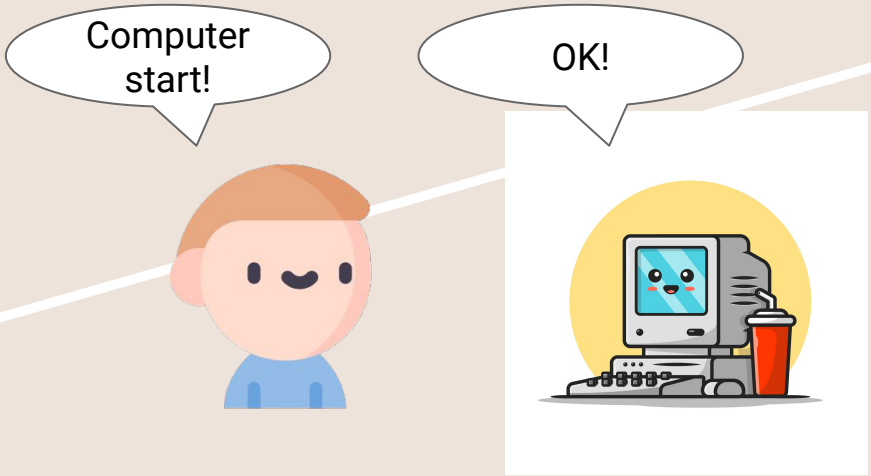






Operativsystem

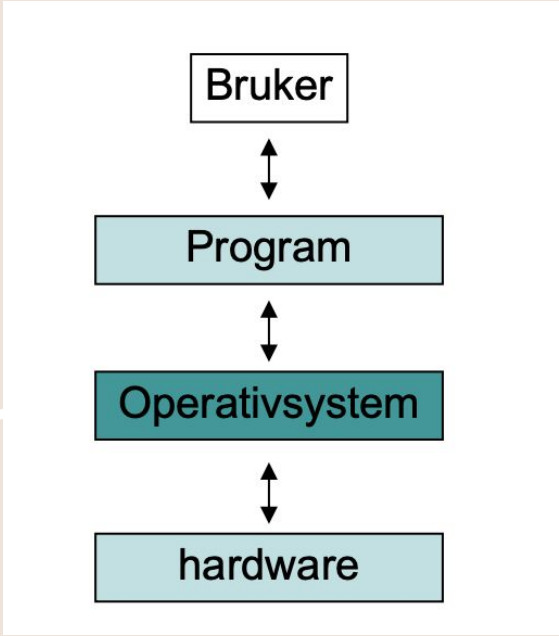
- **Definisjon**: Det er en samling av programmer eller funksjoner som opptrer mellom brukere og hardware. Man skal tilby et høytnivå grensesnitt med et lavnivå hardware (CPU), og samtidig tilby funksjoner for programmer effektivt og sikkert.



Noen oppgaver:

- Administrere kjøringen (starte og stoppe programmer)
- Administrere lagring av filer.
- Første som lastes inn i minnet når maskinen startes, og det styrer kjøringen av alle programmer som lastes inn senere.

To vinkler for å se på operativsystemet



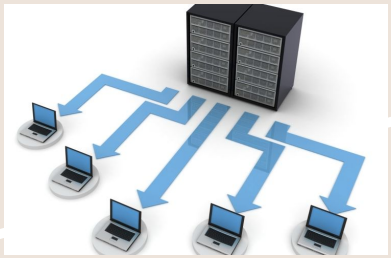
(topp-down)

(bottum-up)

Forskjellige typer operativsystemer (OS)

1. ***Single user, single task***
   a. [redacted]
2. ***Single user, multi-tasking:***
   a. [redacted]
3. ***Multi-user,multi-tasking***
   a. [redacted]
4. ***Distriuted OS***
   a. [redacted]
5. ***Real-time OS***
   a. [redacted]
   b. [redacted]
6. ***Embedded OS***
   a. [redacted]

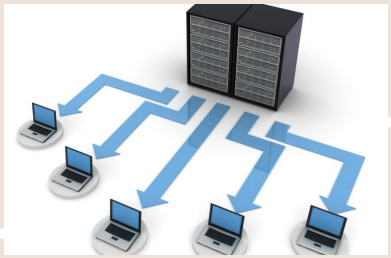






Forskjellige typer operativsystemer (OS)

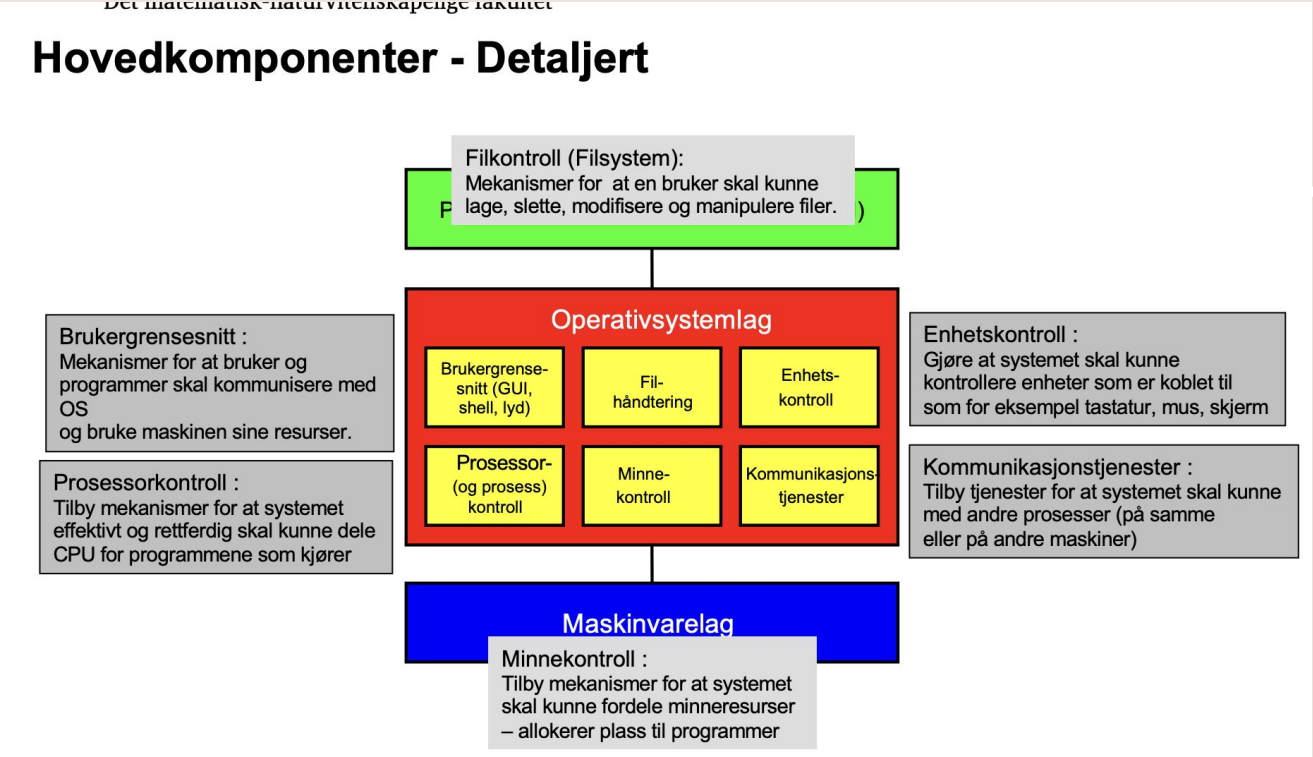






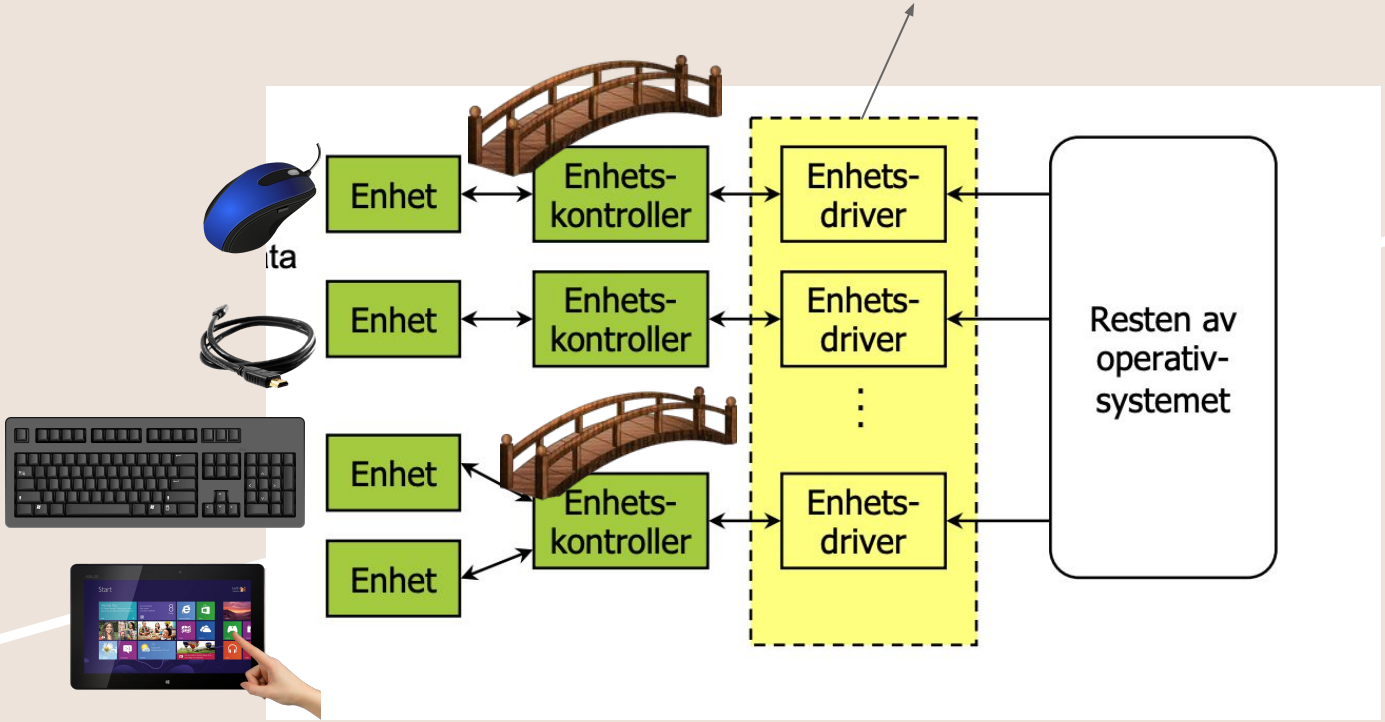
1. ***Single user, single task***
    a. Gamle datamaskiner
2. ***Single user, multi-tasking:***
    a. Smart telefoner
3. ***Multi-user,multi-tasking***
    a. Ifi maskiner
4. ***Distributed OS***
    a. Distribuert ressursfordeling krever et distribuert operativsystem.
5. ***Real-time OS***
    a. Brukes i systemer som krever nøyaktig og umiddelbar respons
    b. Bil
6. ***Embedded OS***
    a. Hvitevarer som vaskemaskiner osv.

Hvordan fungerer operativsystemet?



Enhetskontroll

Software som kommuniserer med device controller og gir kommandoer.

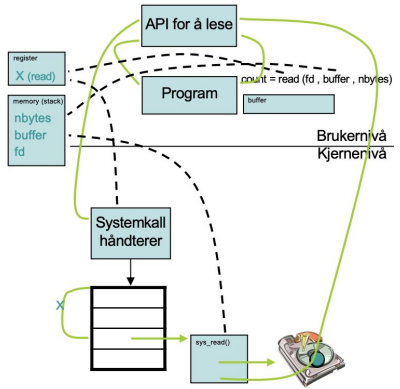


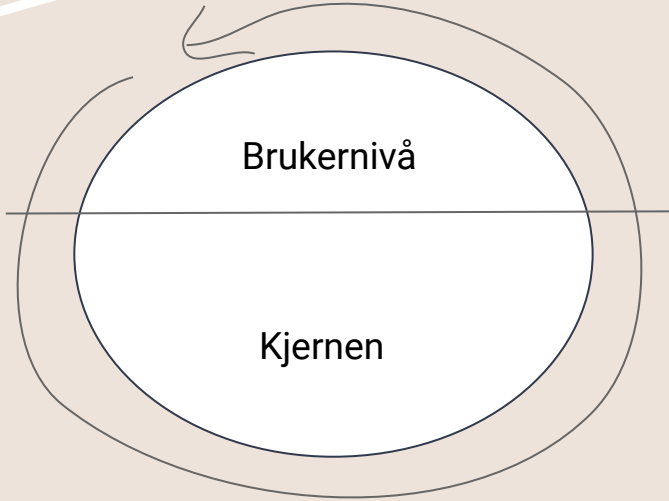
Hvordan fungerer OS-et når man leser en fil?

- Brukergrensesnitt: Koblingen mellom bruker og datamaskinen
- Dersom man skal begynne å gjøre noen operasjoner gjør man det via **systemkall**
  - Går til ***kjernen i OS-et*** for å så hente det opp i applikasjonen.

- Eksempel i programmeringsspråket C:
  `count = read(fd,buffer,nbyte)`

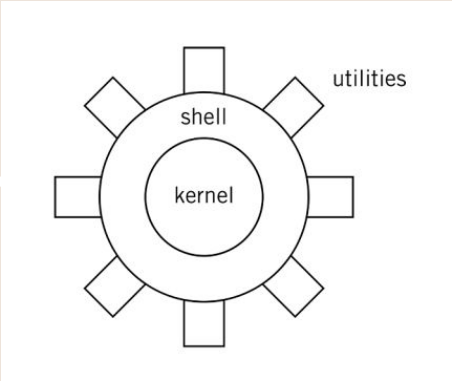
1. Lagrer parametere i minne
2. Kaller API som programmet bruker
3. Setter systemkall nummer i minne
4. Kaller kjernen (TRAP)
   - Kjernen undersøker systemkall nummer
   - Kjernen finner hvor kallet skal sendes
   - Kjernen utfører operasjonen
5. Tilbake til API for å rydde opp
   - Fjerner parametere fra minne
6. Fortsetter programmet

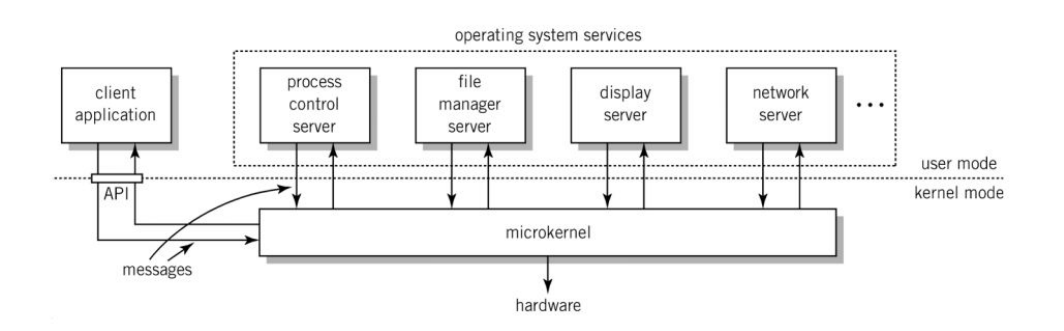




Kjernen i OS-et

- *Den er ansvarlig for å administrere og kontrollere systemressurser som minne, enhetsinnganger, filsystemer…osv.*
  1. **Monolitisk kjerne:** OS kjerne der vi linker alt i et stort program. Her har vi enkel og lett tilgang til alt.
     a. fordel: Systemet blir effektivt.
     b. Utfordring: men om feil oppstår, så kræsjer andre komponenter i systemet
  2. **Mikrokjerner**: ligger på toppen av hardwaren. Alt annet av tjenester bygges som egne komponenter og kjøres ved siden av.





Rettigheter

- En fil er en samling av bytes lagret i eller ved datamaskinen på disk, SSD, HDD eller lignende.
- Operativsystemet holder orden på filens navn, hvilken enhet den er lagret på og hvor i mappestrukturen den ligger.
- Fillagring skjer i et tresystem. Kommandoen ls (list files) forteller oss hvilket filer som er i mappen.

Beskyttelse av filer: unix operer med tre kategorier brukere.

user (forkortet «**u**») er filens eier, dvs den som opprettet den.

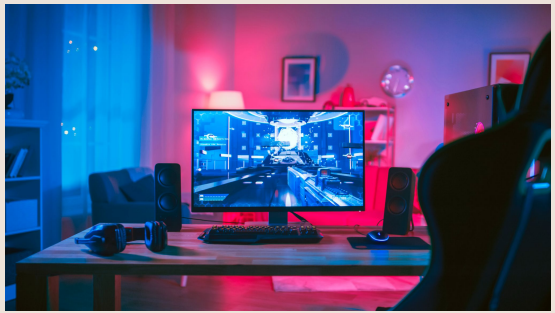
group (forkortet «**g**») er gruppen tilordnet filen. (Mer om dette siden.)

other (forkortet «**o**») er alle andre brukere.

Booting

- Når vi trykker på strømknappen på maskinen vår, så kalles et basic input-output system, en BIOS. Lagret i en slags persistent ram.
- Det vil da si at det gjøres noen sjekker på at ting og tang oppfører seg riktig i det strømmen kommer på.
- Når vi da har fått klar signal fra at de komponentene som er tilkoblet, og responder med en riktig returverdi (passord), så starter man ofte det man kaller bootstrappen.
    - flytte OS-et fra et persistent lagringsmedium til inn i minnet og starte det.





Uke 38 - IN1020 oppgaver

Oppgaver fra Introduksjon til Operativsystemer (IN2140)

September 2023

1 **Hvorfor har vi operativsystemer?**

2 **Hvilke to perspektiver bruker vi for hva et operativsystem er?**

3 **Hvilke hovedkomponenter finner man i et operativsystem? Forklar hva de heter og deres formål.**

4 **Hva er et systemkall?**

5 **Hvilke to typiske kjerne-arkitekturer har vi? Hva er forskjellen?**

6 **Hva er forskjellen på et program og en prosess?**

7 **Hvorfor trenger vi å vite hvordan operativsystemer fungerer?**

Ukesoppgaver + oblighjelp :)