

## 11.3 Operating system design issues

Tuesday, April 28, 2020 9:29 PM

De to viktigste målene for design av I/O-anlegg er effektivitet og generalitet.

Effektivitet er viktig ettersom I/O-operasjoner ofte former en flaskehals i computersystemet. I/O-enheter er mye tregere enn hovedminnet og prosessoren – dette kan løses med multiprogrammering. Likevel klarer uansett ikke I/O-enheter å oppnå samme fart som prosessorer.

Generalitet handler om å behandle enhetene på en uniform måte. Det kan brukes en hierarkisk, modulær metode i designet av I/O-funksjonen. Dette gjemmer mange av detaljene hos I/O til enhetene i lavnivå rutiner, slik at prosesser og øvre lag av OS-et ser på enhetene ved bruk av gjenrelle funksjoner som *read*, *write*, *open*, *close*, *lock* og *unlock*.

### Logisk struktur av I/O-funksjonen

I/O-funksjonen er bygget opp hierarkisk med tre lag av abstraksjon: *logisk I/O*, *enhet-I/O*, og *fysisk I/O* mellom brukerprosessen og hardware.

- **Logisk I/O:** Håndterer de logiske ressursene og er dermed ikke med på den fysiske kontrollen av enheten. Dette inkluderer generelle I/O-funksjoner på vegne av brukerprosesser, som å benytte kommandoer som *open*, *close*, *read* og *write*.
- **Enhet-I/O:** Etterspurte operasjoner og data konverteres til passende I/O-instruksjoner, kommandoer og kontrollordre. Buffere/caching kan brukes for å forbedre utnyttelse.
- **Fysisk I/O (Tidsstyring og kontroll):** Den faktiske tidsstyringen av I/O-operasjonene i tillegg til kontroll av operasjoner skjer her. Dette laget håndterer avbrudd, samt henter og rapporterer I/O-status.

Figuren under viser en modell av I/O-organiseringer:

