

# 8.1 Hardware and control structures

Tuesday, April 21, 2020 1:24 AM

*Se kap 2.3 for kort intro til hva et virtuelt minne er*

Som forklart under kap 2.3 så er det virtuelle minnet avhengig av et samarbeid mellom hardware (i form av prosessor) og software (i form av OS-et) når det opererer minnehåndtering. Under denne seksjonen ser vi nærmere på henholdsvis hardware-aspektet og software-aspektet.

Noen begreper det er kjekt å kunne for videre lesing:

- **Virtuelt minne:** En type lagring hvor sekundærminnet kan adresseres som om det var en del av hovedminnet. Tildelt på disk. Muliggjør effektiv multiprogrammering og frigjør bruker fra begrensninger hos hovedminnet.
- **Virtuell adresse:** Adressen til en lokasjon i det virtuelle minnet som tillater at lokasjonen kan aksesseres som om den var i hovedminnet.
- **Virtuelt adresserom:** Den virtuelle lagringsplassen tilhørende en prosess.
- **Adresserom:** Rekken av minneadresser som er tilgjengelig for en prosess.
- **Reell adresse:** Adressen til en lokasjon (lagringsplass) i hovedminnet.
- **Beboende sett:** Dette er mengden av prosessen som befinner seg i hovedminnet ved et gitt tidspunkt.
- **Reell minne:** Annet navn på hovedminnet, ettersom det er kun her prosesser blir utført.

## Gjennombrudd

Det er to karakteristikker ved enkel paging og enkel segmentering som har vært banebrytende sammenlignet med statisk- og dynamisk partisjonering. Selve gjennombruddet er:

*Dersom de to underliggende punktene er tilstede i systemet, så er det ikke nødvendig at alle pages eller alle segmenter tilhørende en prosess ligger i hovedminnet under execution.*

1. **Dynamisk oversettelse:** Alle minnereferanser i en prosess er logiske adresser som dynamisk oversettes til fysiske adresser ved kjøretid. Dermed kan en prosess byttes inn og ut av hovedminnet slik at den opptar ulike områder av hovedminnet under ulike tidspunkt av execution.
2. **Oppdeling av prosesser:** En prosess kan deles inn i flere deler (pages eller segmenter) som ikke trenger å ligge sammenhengende i minnet under kjøringen. Det er den dynamiske oversettelsen og bruk av page-/segment-tabell som muliggjør dette.

Dette vil si at en prosess kan fortsette å kjøre så lenge delen som holder neste instruksjon, og dens tilhørende data som skal aksesseres, ligger i hovedminnet. Dersom execution av en del av prosessen krever en del som ligger i sekundærminnet vil prosessen blokkeres av OS-et, som så gjør en I/O-forespørsel for å bytte inn blokken med referansen. I mellomtiden kan en annen prosess kjøre.

Det er to fordeler med denne strategien som forbedrer systemets utnyttelse:

1. **Flere prosesser kan oppta hovedminnet.** Fordi vi kun laster inn deler av prosessene vil det være mer plass til flere ulike prosesser.
2. **En prosess kan være større enn hele hovedminnet.** Denne egenskapen er utrolig kraftig. Ved bruk av det virtuelle minnet vil man kunne kjøre programmer som i utgangspunktet ville krevd for mye minne, men som ved utnyttelsen av virtuelt minne med paging eller segmentering ikke er noe problem. Denne jobben håndteres av hardware og OS-et, altså trenger ikke programmerer å bekymre seg for størrelsen på minnet.

*Sidenote: En computer kan ha 8 GB RAM, som altså er størrelsen på hovedminnet. Likevel kan en computer kjøre programmer med prosesser som betydelig overgår disse 8 GB – og dette er nettopp fordi computeren kan med sitt virtuelle minne bruke disk, altså plass fra sekundærminnet til å holde på deler av prosessene underveis i kjøringen. Bare å sjekke med aktivitetsmonitor på Mac.*

Det virtuelle minnet baseres i grunn på operativsystemets evne til å bytte ut prosesser fra hovedminnet og sekundærminnet. Likevel kan det oppstå problemer her, da det er mulig at OS-et bytter ut en prosess som snart skal brukes. Dette kan bli særdeles ineffektivt; og denne tilstanden kalles *thrashing* – hvor OS-et bytter frem og tilbake såpass mye at det sinker systemet. Dette problemet omfattes i konseptet om lokalitetsprinsippet (Se kap 1). Dermed har det blitt utarbeidet effektive algoritmer som tar valg basert på gode gjetninger.

