1.5 The memory hierarchy

Sunday, March 22, 2020 5:22 PM

Det er tre grunnleggende karakteristikker ved minne: kapasitet, aksesstid, og kostnad.

Det er en trade-off ved hver av disse karakteristikkene. De følgende relasjonene gjelder for hver karakteristikk:

- Raskere aksesstid, større kostnad per bit
- Større kapasitet, mindre kostnad per bit
- Større kapasitet, mindre aksesstid

Løsningen til designeren på dette problemet følger et minnehierarki (nedover):

a. Synkende kostnad per bit
b. Større kapasitet
c. Stigende aksesstid
d. Synkende frekvens av aksess til minne av prosessoren

Def. *Hit ratio (H)*	Brøkdelen av alle minneaksesser som er funnet i cache

Eks. Anta en prosessor har to nivåer av minne. Nivå 1 inneholder 1000 bytes og har en aksesstid på 0.1 µs. Nivå 2 inneholder 100,000 bytes og har en aksesstid på 1 µs. Anta at 95% av minneaksessene er i cache (H = 0.95). Da er den gjenomsnittlige aksesstiden på en byte gitt ved

(0.95)*(0.1 µs) + (0.05)*(0.1 µs + 1 µs) = 0.15 µs

Resultatet er nærme aksesstiden på det raskere minne (cachen), så strategien med to nivåer fungerer greit prinsipielt, men bare hvis betingelsene (a) til og med (d) i hierarkiet er tilstede. Basisen for om (d) er valid er et prinsipp generelt kalt lokalitetsprinsippet (locality of reference).

Def. *Locality of reference*	Ved execution av et program refererer prosessoren til data og instruksjoner i minne, og de har en tendes til å klynges (cluster). Over tid vil disse klyngene endres, men I en kort tidsperiode jobber prosessoren hovedsaklig med gitte klynger av minnereferanser. Dette kan utnyttes ved å organisere data på tvers av hierarkiet slik at prosentandelen av aksesser nedover langs hierarkiet blir mye mindre for hvert steg ned.

Det er tre hovedformer av minne: inboard memory, outboard storage, off-line storage:

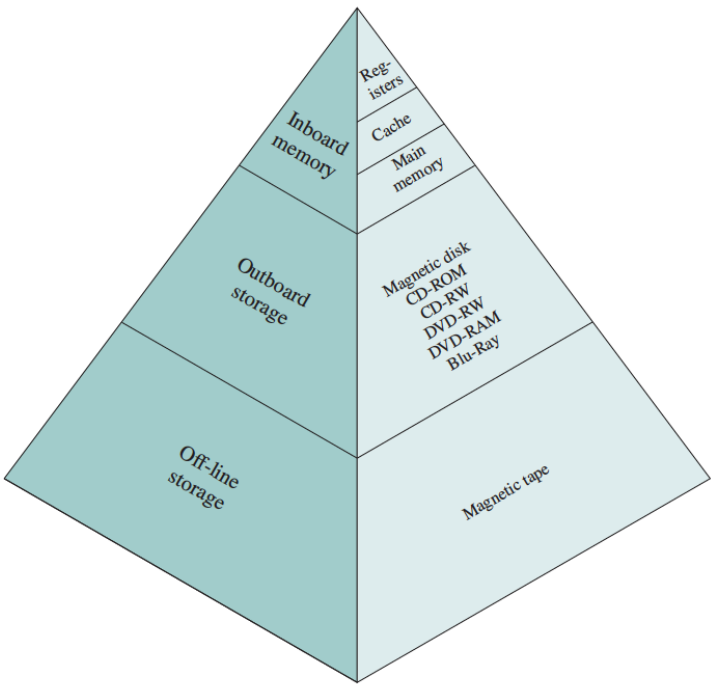

Figure 1.14 The Memory Hierarchy

Det kan legges til flere lag i hierarkiet i software. En del av hovedminne kan eksempelvis brukes som et buffer til å midlertidig holde på data som skal leses fra disk. En slik teknikk, kalt disk cache

kommer vi tilbake til I Kap 11, men forbedrer ytelsen på to måter:

1. Disk writes blir klynget. Istedenfor mange, små overføringer av data har vi få, store overføringer av data. Dette forbedrer disk ytelsen og minimerer prosessores deltakelse.
2. Noe data som er ment for write-out kan bli referert av et program før neste dump til dusk. I så fall, dataen er motatt hyppig fra software cache istedenfor sakte fra disk.