

TK1100

FORELESNING 0x00 INTRO

Noen kjappe tips til oppgavene:

- Skriv med dine egne ord, oppsøk gjerne andre kilder enn forelesningen som kan hjelpe forståelsen din.
- Hold det til maks. 5 – 6 setninger på spørsmålene der det står «Forklar kort». Andre spørsmål kan det være fint å utdype grundigere.
- Enkelte av oppgavene kan gå litt utenom det som er presentert i forelesningene, bruk gjerne veilederene til å peke deg i riktig retning!

1) Datamaskinens bruksområder

- a) Forklar kort hva et PC-system er.

Et PC-system er et system der ytelsen er å transportere data og instruksjoner mellom komponentene som finnes i PC'n og få de til at bli prosessert raskt. Her er det viktig at komponentene er i balanse med hverandre så at de ikke holder hverandre tilbake, de burde med andre ord være på samme nivå.

- b) Gjør din egen definisjon av hva «informasjon» er i datasammenheng.

Her finnes det flere svar, det viktigste er å skjønne at «informasjon» kan bety flere ulike ting i ulike sammenhenger.

- c) Nevn noen bruksområder for datamaskiner. Bruk gjerne eksempler fra din egen hverdag. Hva er de viktigste hovedmomentene ved datamaskinene du har i din egen hverdag?

Datamaskiner finnes i dag overalt, her vil vi at du reflekterer over hvilke datamaskiner som finnes i din hverdag og hva bruksområdene deres er.

Eksempel på hovedmomenter med datamaskiner:

- Hastighet
- Pålitelighet
- Lagringsevne
- Pris
- Størrelse

Eksempel på bruksområder for datamaskiner:

- Embedded systems (Trafikklys, kaffetrakter etc.)
- Mobiltelefon
- Datalagring
- System til banker, kommune etc.
- Forskning

d) Forklar med egne ord forskjellene mellom det binære tallsystemet og titall-systemet.

Bruk gjerne tegninger for å vise hva du mener.

Her bør svaret ditt ha vært innom:

- Titalls-systemet er det vi bruker i hverdagen i Norge, og det vi lærer mest om på skolen.
- Det binære tallsystemet bruker det samme konseptet med «plasser» som titallssystemet, men representerer andre verdier ved hver plass:

Når vi skriver tallet 10 i titallssystemet, betyr dette egentlig at vi har:

1 x tierplassen og 0 x enerplassen som gir oss tallet ti.

Når vi skriver tallet 10 i det binære tallsystemet, betyr dette egentlig at vi har:

1 x toerplassen og 0 x enerplassen som gir oss tallet to.

- Base betyr hvilket tallsystem vi oppgir et tall på.

Base 10 refererer til titallssystemet og *base 2* det binære tallsystemet, men vi er ikke begrenset til å oppgi tall på bare disse to basene.

- Totalls-system brukes for å «snakke» med datamaskiner.
- Et binært siffer kalles en bit
- En gruppe med 8 bit kalles for en **byte**

e) Skriv ned tallene 1 – 15 på det binære tallsystemet.

| | | | |
|------|------|------|------|
| 0 | 01 | 10 | 11 |
| 100 | 101 | 110 | 111 |
| 1000 | 1001 | 1010 | 1011 |
| 1100 | 1101 | 1110 | 1111 |

f) Hvorfor bruker datamaskiner det binære tallsystemet og ikke titalls-systemet?

Forenklet forklart fungerer en datamaskin ved at strøm sendes rundt omkring som signaler i maskinen med beskjed om hva som skal utføres. De ulike komponentene i maskinen tolker hva de skal gjøre basert på signalet de får inn. Det kan være forvirrende å skjønne hvordan dette knyttes opp mot det binære tallsystemet, men det kan være lurt å se på hva som ville skjedd hvis datamaskiner hadde brukt titallsystemet i stedet.

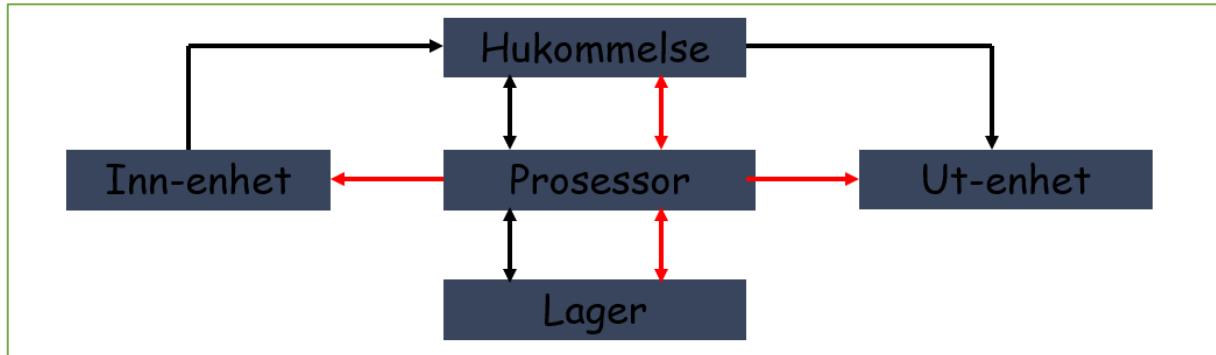
La oss se for oss et scenario der datamaskiner bruker titallssystemet og med det har **10** mulige ting som kan skje basert på et innkommende signal. Vi måler signalet i maskinen til å være 2,5 på en 0-9 skala. Skal vi tolke den målingen som 2 eller 3? Her blir det vanskelig for maskinen å *tolke* signalet, og skjønne hva som skal gjøres videre.

Gitt det samme scenarioet der datamaskiner i steden bruker *det binære tallsystemet*. Hvis vi bare har **to** mulige ting som kan skje kan vi jo si at alle målinger mellom 0 – 4 er **0** og alle målinger mellom 5 – 9 er **1**. Med den samme målingen på 2,5 kan vi være helt sikre på at signalet egentlig skal tolkes som en 0.

Vi bruker altså det binære tallsystemet med datamaskiner blant annet fordi det eliminerer usikkerheter. Datamaskinen må vite nøyaktig hva den skal gjøre når den får et innkommende signal.

2) Datamaskinens bestanddeler

- a) Tegn opp Von-Neumann modellen og delene den består av.



- b) Hvor mange lag har en funksjonsorientert modell mot en maskinvareorientert modell?

En funksjonsorientert modell har totalt 6 lag og disse er:

- Lag 5: Brukerprogramnivå
- Lag 4: Kompilatornivå/interpreter
- Lag 3: Operativsystemnivå
- Lag 2: Instruksjonsnivå
- Lag 1: Mikroinstruksjonsnivå
- Lag 0: Digital kretsnivå

En maskinvareorientert modell har totalt 4 lag og disse er:

- Portnivå
- Registrernivå
- Prosessornivå

- c) Gi et par eksempler på hardware.

Hardware/maskinvare i datasammenheng er fysiske komponenter som håndterer prosessering, datalagring og inn-/ut-operasjoner.

Eksempel:

- Harddisk
- RAM
- CPU/prosessor
- GPU
- CD-ROM
- Diskett
- Mus
- Skriver

d) Gi et par eksempler på software.

Software/programvare er en «abstrakte» eller ikke-fysiske applikasjoner som kjøres på en datamaskin, disse er oppbygd med et programmeringsspråk som datamaskinen skjønner.

Eksempel:

- Skype, Zoom, Slack
- Word, Excel, PowerPoint
- Chrome, Safari, Edge

e) Hvorfor kalles «hjernen» i en datamaskin CPU eller «Central Processing Unit», og hvordan gir man den instruksjoner?

Det er CPU-en som utfører alle instruksjoner vi gir til datamaskinen. Når vi vil at datamaskinen skal gjøre noe må det alltid «gjennom» CPU-en. Den kan derfor tenkes på som en hovedmotor. Programmeringspråk gir oss mulighet til å sende instruksjoner til en prosessoren.

f) Skriv om tre valgfrie typer av minne. Bruk gjerne andre kilder enn forelesningene for å finne informasjon.

Eksempel på minne som kan skrives om:

- RAM
- SRAM
- DRAM
- ROM
- FLASH
- Virtuelt minne
- CACHE

g) Hva er forskjell på minne og masselager?

Et masselager er der data er lagret permanent, masselager brukes til alt fra f.eks. dine personlige bilder, til filer datamaskinen trenger for å fungere på lang sikt. Minne er der data ligger tilfeldig for at den skal brukes av prosessoren i datamaskinen. For eksempel så er et program lagret på din harddisk(masselager), men når du kjører programmet så er deler av programmet i RAM(minne) hvor det tar mye kortere tid for prosessoren å hente informasjon fra. For å overføre data til prosessoren fra masselager så brukes det en kontroller, en kontroller kan være IDE/EIDE, SATA eller SCSI.

Eksempel på masselager:

- CD-ROM
- USB
- Hullkort
- Harddisk
- Diskett
- Magnetbånd

h) Prøv å finne info om komponentene på din egen datamaskin. Eks. Hvor mye plass har du på din harddisk eller hvilken type CPU har du?

Dette er en praktisk oppgave som har ulike svar basert på hvilken datamaskin du har, for å finne ut av hva svaret er så er det enkleste å google hvordan du skal gå frem.

- i) Forklar hva et abstraksjonsnivå er, og hensikten med det.

Et abstraksjonsnivå er et begrep på at man gjemmer bort kompleksitet for å forenkle hvordan man forstår og forholder seg til maskin- eller programvare. Jo *høyere* abstraksjonsnivå, jo *færre* detaljer og kompleksitet forholder man seg til. Jo *lavere* abstraksjonsnivå, jo *mer* detaljer og kompleksitet forholder man seg til.

Om du for eksempel skal utvikle en nettside er det sjeldent du trenger å ha et forhold til hvordan en prosessor fungerer ned på et fysisk nivå (f.eks. hvordan strømmen beveger seg i datamaskinen). Men det kan likevel være veldig fint å ha en konseptuell forståelse av hva en prosessors rolle er i en datamaskin og hvordan den oppnår noe av det. Da kan man si at denne personen forstår en prosessor på et *høyt* abstraksjonsnivå.

- j) Forklar kort hva et operativsystem er.

Operativsystemet er programvare som gir et brukergrensesnitt, sikkerhet og pålitelighet. OS-et gjør det mulig å kjøre de fleste applikasjoner og administrerer ressursene som finnes i datamaskinen gjennom å tildele disse til andre programmer. En viktig funksjon operativsystemet har er at det ser til at flere programmer ikke kan sende data til samme ressurs samtidig, men legger ting i venteliste. Det kan også være verdt å nevne at alle datamaskiner er ikke *må* ha operativsystem.

3) Passord

- a) Forklar kort hva bitstyrke er og hvorfor det er viktig i passordsammenheng

Bitstyrke sier noe om hvor sterkt et passord er. For å måle dette ser man på hvor mange forsøk en tilfeldig angriper maksimalt trenger for å gjette passordet. Formelen for å regne ut bitstyrken er:

$$\lg_2(\text{Antall } \textit{mulige} \text{ karakterer}) * \text{antall karakterer}$$

Gitt et passord på **4** karakterer der det kun er lov med norske bokstaver, altså **29** *mulige* karakterer, vil utregningen se slik ut:

$$\lg_2(29) * 4 =$$

$$4,86 * 4 =$$

$$\underline{\underline{19,44}}$$

Anbefalt bitstyrke er på ca. 80. Dette kan da være en hvilken som helst kombinasjon av karakterer og mulige karakterer.

- b) Hva er brute-force?

Brute-force er en metode for å finne/gjette riktig passord gjennom å prøve alle mulige kombinasjoner som finnes.

c) Er et passord som inneholder åtte bokstaver nødvendigvis sikkert?

Nei det er ikke nødvendigvis sikkert.

Når vi regner ut bitstyrken til et passord ser vi på antall mulige karakterer i tillegg til lengden på passordet. I denne oppgaven står det bare «bokstaver», men *hvilke* bokstaver snakker vi om?

Er det bare lov med bokstaver fra det norske alfabetet i passordet vil det være 29 mulige karakterer. Det vil være sterkere enn f.eks. det engelske alfabetet som bare har 26 bokstaver og dermed 26 mulige karakterer. For å avgjøre om et passord er sikkert er det derfor viktig at vi vet omstendighetene rundt passordet.

En annen faktor vi kan trekke inn er om de 8 bokstavene er del av et vanlig passord. Brute forcing er sjeldent tilfeldig, og man prøver gjerne de mest åpenbare mulighetene først. Gitt at de 8 bokstavene i passordet er «Password» kan man derfor regne med at passordet ikke er særlig trygt.

d) Hva er tiltak man kan gjøre for at et passord blir sterkere?

- Bruke mer enn bokstaver og tall, eksempel spesialtegn som: @£\$€{[]}
- Ha blanding mellom store og små bokstaver
- Ikke bruke samme passord på flere ulike tjenester
- Lagre passordet sikkert

e) Sorter passordene under etter styrke, start med den som er minst sikker. Bruk f.eks.

Microsoft sin passordsjekker til å teste svarene dine

- iliketoast
- qwerty
- password123
- jegelskerågåturiskogenpålørddager

Med utgangspunkt i at alle passordene kan ha tall fra 0-9 og norske karakterer vil styrken rangert fra svak til høy være følgende:

qwerty

iliketoast
password123
jegelskerågåturiskogenpålørddager

4) Historie

- a) Skriv om de tre historiske problemstillingene den moderne datamaskinen løser.

Den moderne datamaskinen løser hovedsaklig tre ulike problemstillinger:

- Beregningsproblemet. Altså hvordan kan vi utføre kompliserte beregninger på en rask og sikker måte. Her kan man se for seg at man tidligere satt og gjorde komplekse beregninger med kalkulator og papir og penn. Tenk på alle feilkildene som kan oppstå når et menneske skal utføre det mot en dagens datamaskin.
- Massedataproblemet. Altså hvordan man kan lagre store mengder data på en pålitelig måte og behandle den. Tidligere brukte man kanskje papirarkiv som både er upålitelig og vanskelig å behandle effektivt.
- Reguleringsproblemet. Altså hvordan kan man styre og automatisere industrielle prosesser? Før hadde man mange fler industrielle arbeidere som styrtede ulike deler av f.eks. en produksjon. I dag er mye helautomatisert, eller styres fra en eller flere datamaskiner.

- b) Hva er Moore's lov og hva beskriver den? Vil den for alltid være sann?

Moore's lov er en observasjon (og teknisk sett ikke en lov), der man har sett at antall transistorer på et areal vil dobles hver 24. måned. Transistorer kan beskrives som det som får en datamaskin til å fungere, og man kan forstå det som at *flere* transistorer vil gi *høyere* ytelse i en datamaskin. Moore's lov beskriver derfor farten på den teknologiske utviklingen vi har sett siden midten av forrige århundre.

Det er ingen enighet om når Moore's lov kommer til «å dø», eller med andre ord slutte å være sann. Men det kommer jevnlig uttalelser fra ulike folk i industrien som konkluderer med at den allerede er død og at vi dermed kommer til å se en stagnert utvikling i antall transistorer/areal i fremtiden.

- c) Hva er Intel 4004, og hvilken generasjon hørte den til?

Intel 4004 er en 4. generasjons 4 bit CPU produsert av Intel, lansert i 1971. Den var både den første CPU-en produsert av Intel og den første *mikroprosessoren* som ble kommersielt tilgjengelig. En mikroprosessor mer spesifikt er en prosessor som kun består av én eller veldig få integrerte kretser.

- d) Hva beskriver Wirth's lov?

Wirth's lov er som Moore's lov altså heller ikke teknisk sett en lov, men et slags ordtak eller en «saying» som konstaterer at programvare blir tregere forttere enn maskinvare blir raskere.

5) Binær koding

- a) Trening til neste økt, fortsett å skrive ned tall på det binære tallsystemet opp til 128: 1, 2, 4, 8...