

Fra bits til nyttig informasjon og datamaskinens komponenter

ARBEIDSKRAV EMNE 3 – DIGITAL TEKNOLOGI



Student: Christian Westbye

[Github Repo.](#)

Oppgave 1

Musikkindustri: Fra fysisk til digital modell.

Den digitale revolusjonen har forandret hvordan musikk benyttes, distribueres og handles. Dette har skjedd svært raskt i løpet av de siste 20-30 år, med et klart skifte fra fysiske til digitale format.

Man hadde vinyl, kassett og til slutt CDer på 90-tallet, en stabil høy pris og en stor andel samlere og kjøpere. Musikkbutikker fantes overalt, både i små og store byer. Musikkindustrien solgte godt og brukte store ressurser på artister og produksjoner for å posisjonere seg i markedet. Distribusjonsavtaler var svært viktige og avgjorde ofte hva du fant i din lokale musikkbutikk – både kjeder og selvstendige butikker var vanlige.

Mot slutten av 1990-tallet koblet internett verden sammen på en helt ny måte. Plutselig trengte man ikke lenger kjøpe musikkblader fra utlandet for å oppdage nye artister, og mulighetene økte for å finne musikk utenfor de store distribusjonsnettverkene.

Med ny kraftig komprimering av musikk filer som kom for fullt på slutten av 90-tallet (mp3) kunne man plutselig laste ned musikk som før var for tung (full audio cd 74minutt, 700MB) og tidkrevende for modem og isdn linjer (14.4kilobits per sekund eller ISDN med 64kilobits per sekund). Med en mp3 med 64kbs komprimering kunne et 45minutt album komme ned på 20MB.

Dette førte også til CD-brenning og kopiering av mange album/sanger på én enkelt CD. Dedikerte MP3-spillere, som for eksempel iPod, tok trenden videre ved å la folk ha med seg hele musikkbiblioteket i lomma.

Rundt årtusenskiftet og utover 2000-tallet pågikk det mange rettssaker for å stoppe fildelingen. Flere tjenester ble stengt ned, men kopieringen fortsatte likevel – enten via nye tjenester eller bare ved at venner kopierte fra venner lokalt.

Internett ble raskere og raskere, og flere selskaper begynte å eksperimentere med strømming av lyd og video direkte, uten at nedlasting var nødvendig.

I 2006 ble Spotify grunnlagt av to svensker som ville tilby musikk på en lovlige måte som både plateselskaper og lyttere kunne være fornøyde med. Løsningen ble en abonnementsmodell der en fast månedspris ga tilgang til å strømme (ikke laste ned) all tilgjengelig musikk på tjenesten.

Denne modellen ble også brukt av mange andre tilbydere, kanskje mest kjent av Netflix for film og serier.

Artister, musikere og plateselskaper (de som sitter på rettighetene) får på Spotify og lignende tjenester betalt per avspilling – og summen varierer avhengig av hvilket land lytten er i. Kontroversen rundt små utbetalinger til mindre artister og enorme summer til de med millioner av avspillinger har pågått i mange år, uten at modellen har blitt endret.

I de seneste årene har også fysiske format fått et comeback, spesielt vinyl og til dels kassetter men i et mye mindre volum enn tidligere storhetstider. De selges ofte direkte fra artister eller plateselskaper, siden det er få musikkbutikker og distribusjonen er begrenset.

Digital strømming er i 2025 normalen på lytting av musikk, og mange bruker bare telefon koblet på nett til å strømme direkte fra sin tilbyder.

Oppgave 2

(a)

Forklaring:

bit - Minste verdi for en datamaskin, kan kun være 0 eller 1.

byte - Verdi satt sammen av 8 bit (0 - 7). Med en byte kan vi sette et tegn eller nummer.

(b)

Binærverditabell, totalsystem (Grunntall = 2):

Binær diagtabell som viser posisjon og verdi i hver bit i et 16-bit binært tall, fra høyre til venstre:

Bit Posisjon	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Decimal Verdi	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Utrekning fra desimal til binær

(A) 12

Deling på 2 nedover: uten rest = 0, rest = 1. Utrekning: | 12 / 2 = 0 (ingen rest = 6) | | 6 / 2 = 0 (ingen rest = 3) | | 3 / 2 = 1 (rest = 1) | | 1 / 2 = 1 (rest = 1) |

12 desimal = 1100 binær

Posisjon	3	2	1	0
Desimalverdi	8	4	2	1
Binærtall	1	1	0	0

$$\text{Desimalverdi } (8 * 1) + (4 * 1) + (2 * 0) + (1 * 0) = 12$$

(B) 27

Deling på 2 nedover: uten rest = 0, rest = 1. | 27 / 2 = 1 (rest = 13) || 13 / 2 = 1 (rest = 6) || 6 / 2 = 0 (ingen rest = 3) || 3 / 2 = 1 (rest = 1 = 1) || 1 / 2 = 1 (rest = 1 = 1) |

27 desimal = 11011 binær

Posisjon	4	3	2	1	0
Desimalverdi	16	8	4	2	1
Binærtall	1	1	0	1	1

$$\text{Desimalverdi } (16 * 1) + (8 * 1) + (4 * 0) + (2 * 1) + (1 * 1) = 27$$

(c) Konverter fra binær til desimal

binær = 101110

Posisjon	5	4	3	2	1	0
Desimalverdi	32	16	8	4	2	1
Binærtall	1	0	1	1	1	0

$$\text{Desimalverdi } (32 * 1) + (16 * 0) + (8 * 1) + (4 * 1) + (2 * 1) + (1 * 0) = 46$$

Dobbel hver verdi fra venstre posisjon og nedover, legg sammen til slutt på høyre side for desimaltall

Begynn fra venster: 1 Dobbel 1 = 2 legg til neste verdi , 0 = 2 Dobbel 2 = 4 legg til neste verdi , 1 = 5 Dobbel 5 = 10 legg til neste verdi , 1 = 11 Dobbel 11 = 22 legg til neste verdi , 1 = 23 dobbel 23 = 46 legg til neste verdi , 0 = 46

binær 10111 = 46 desimal

Andre utregning eksempler:

Andre eksempler: 1010 Begynn fra venstre: 1 1 * 2 2 + 0 = 2 2 * 2 4 + 1 = 5 5 * 2 10 + 0 = 10

Binær 0111 = 10 desimal

Begynn fra venstre: 0 0 2 + neste verdi 1 = 1 1 2 + neste verdi 1 = 3 3*2 + neste verdi 1 = 7

Binær 0111 = 7 desimal

(d) Konverter teksten "DATA" fra ASCII til binær:

01000100 01000001 01010100 01000001

Kalkulerings:

Ascii verdien for D er 68.

```
68 / 2 = 34 ingen rest = 0  
34 / 2 = 17 ingen rest = 0  
17 / 2 = 8, rest =      1  
8/2 = 4 ingen rest =      0  
4/2 = 2ingen rest =      0  
2/2 = 1 ingen rest =      0  
1/2 = 0 rest =          1  
0/2 = 0 ingen rest =      0
```

01000100 = D

Ascii verdien for A er 65.

```
65 / 2 = 32 + rest      = 1  
32 / 2 = 16 ingen rest = 0  
16 / 2 = 8 ingen reset = 0  
8 / 2 = 4 ingen rest =  0  
4 / 2 = 2 ingen rest =  0  
2 / 2 = 1 ingen rest =  0  
1 / 2 = 0 + rest =      1  
0 / 2 = 0 ingen rest =  0
```

01000001 = A

Ascii verdien for T er 84.

```
84 / 2 = 42 ingen rest = 0  
42 / 2 = 21 ingen rest = 0  
21 / 2 = 10 rest      =  1  
10 / 2 = 5 ingen rest =  0  
5 / 2 = 2 rest        =  1  
2 / 2 = 1 rest        =  0  
1 / 2 = 0 rest        =  1  
0 / 2 = 0 rest        =  0
```

01010100 = T

Ascii verdien for A er 65.

```
65 / 2 = 32 + rest      = 1  
32 / 2 = 16 ingen rest = 0
```

```
16 / 2 = 8 ingen reset = 0  
8 / 2 = 4 ingen rest = 0  
4 / 2 = 2 ingen rest = 0  
2 / 2 = 1 ingen rest = 0  
1 / 2 = 0 + rest = 1  
0 / 2 = 0 ingen rest = 0  
  
01000001 = A
```

Oppgave 3

(a) Hva er OSI Modellen:

OSI-modellen (Open Systems Interconnection model) er et standardisert (teoretisk) rammeverk som forteller hvordan data-kommunikasjon fungerer i et nettverk. Består av 7 lag, der hver del har sin oppgave og kommuniserer kun med det laget som er over og under seg for å sørge for korrekt data flyt.

(b) Beskriv Lag 2 - Datalinklaget og Lag 3 - Nettverkslaget.

Lag 3: Nettverkslaget (Network Layer) IP-adresser, ruting

Dette er laget der IP-adresser kommer inn i bildet.

Ansvarlig for at datapakker blir levert til riktig destinasjon på tvers av nettverk.

Lag 2: Datalinklaget (Data Link Layer) MAC-adresser, rammer (frames)

Fokuserer på hvordan data pakkes inn i rammer og sendes over en enkelt fysisk forbindelse (f.eks. innenfor samme lokale nettverk).

(c) Forklar hvordan OSI modellen kan hjelpe teknikere å feilsøke nettverksproblem

OSI modellen deler kommunikasjonen inn i 7 lag, hver med definerte oppgaver. Hvert lag kommuniserer kun med laget rett over og under seg som igjen gjør det enklere å forstå og feilsøke nettverks problemer og se hvor feilen har oppstått. Ved å jobbe seg oppover/nedover lagene og teste underveis, kan man lettere finne årsåkan og finne hvor problemet ligger.

Oppgav 4

(a) Forklar RAM og ytelse, hva skjer når minnet er fullt.

Ram (Random Access Memory) er datamaskinens minne som brukes etter behov/setup fra operativsystem og applikasjoner. Det er ett høyhastighets område for lagring av data som maskinen aktivt bruker. Rask tilgang, flytting og prosesering av store mengder data blir da mulig.

Ram kjører på mye høyere lese/skrive hastighet mot en harddisk/sdd, jo mere ram du har jo mer data kan kjøres direkte fra minnet og mindre deler hentes opp fra harddisk/ssd.

Er minnet fullt blir deler av harddisk/ssd brukt som minne (virtual memory). Harddisk/ssd er ikke like rask som minnet og applikasjoner/operativsystem vil da ikke ha samme ytelse og hastighet som om man hadde nok fysisk minne ledig. Denne prosessen blir kalt swapping eller paging.

Har man ingen diskplass og ram igjen så kan man oppleve at systemet står/fryser og/eller applikasjon kan avslutte eller krasje for å så å frigjøre minne (henger applikasjonen kan man måtte avslutte manuelt for å frigi

minnet).

(b) Oppgradere ram men eldre CPU, hvilke flaskehals er kan oppstå.

Man kan oppleve at man ikke får utnyttet hastigheten på ram brikker viss minne kontrolleren i denne eldre cpu ikke støtter denne høyere hastighet, man må også passe på å kjøpe en rett ram type til sitt hovedkort. Har man opplevd at minnet ble fullt ofte før oppgradering av minnet vil man oppleve at maskinen er raskere siden man vil slippe å bruke for mye virtuell minne og harddisk/ssd som må flytte på data.

Selv regneoperasjoner vil ikke få høyere hastighet i seg selv med en slik oppgradering (ny cpu trengs da).
