

## Sammenligning av 2- og 10-tallsystemene

Totallsystemet	Titallsystemet	Toerplasser	Toerpotens
0	0		
1	<b>1</b>	1 toerplass	$2^0$
10	<b>2</b>	2 toerplasser	$2^1$
11	3		
100	4	3 toerplasser	$2^2$
101	5		
110	6		
111	7		
1 000	<b>8</b>	4 toerplasser	$2^3$
10 000	<b>16</b>	5 toerplasser	$2^4$
100 000	<b>32</b>	6 toerplasser	$2^5$
1 000 000	<b>64</b>	7 toerplasser	$2^6$
10 000 000	<b>128</b>	8 toerplasser	$2^7$
100 000 000	<b>256</b>	9 toerplasser	$2^8$

## Bits og bytes i datamaskinene

«Begrepet ble først benyttet av [Claude E. Shannon](#) i en artikkel fra [1948](#). Ordet er en sammentrekning av *binary digit* (eventuelt *binary digit*).»<sup>1</sup>

Med 8 bits per byte kan du gjengi 256 verdier fra 0 to 255, som vist her, i forlengelsen av tabellen ovenfor:

$$254 = 11\ 111\ 110$$

$$255 = 11\ 111\ 111$$

I datamaskinenes barndom fastsatte man en byte til 8 bits, bl.a. fordi det er så mye man trengte for å vise nok antall fargenyanser (256 ulike er omtrent så mye som et menneskeøye kan skjelve) og til å gjengi ett tegn (bokstaver, tall o.l.).<sup>23</sup>

Dermed lagde man vanligvis 8 bits-maskiner, altså maskiner som kunne overføre 8 bits i slengen (det kalles på dataspråket å adressere 8 bits minne). Senere har dette blitt økt først til 16, så 32 og (ca. 2003) til 64. Våre dagers datamaskiner foretrekker altså å bunte sammen 64 og 64 ledninger der informasjon skal overføres, slik at det gir  $2^{64}$  antall muligheter for overføringer. Det er nyttig for ekstremt store tall, men gir ellers ikke raskere datamaskiner om det er små tall som like godt kunne vært håndtert av f.eks. En 16-bits datamaskin. Med unntak av om du har mange programmer åpne samtidig, og/eller håndterer (flere) veldig store filer på en gang. For 64-bits maskiner har mulighet for mer arbeidsminne (RAM).

## Minnebegrensninger

Minnebegrensningen til 32-bits programvare kan forklares med litt matematikk. Et 32-bits system har maksimalt tilgang til ca. 4,2 millioner adresser – som tilsvarer det 4 GB minne kan tilby.<sup>4</sup> 4,2 millioner er  $2^{32}$ . Med et 64-bit systemer får  $2^{64}$ . Det gir 17,2 milliarder gigabyte med minne (16 exabyte). På samme måte som da man gikk fra 16 til 32 bit systemer, tror man også i dag et 64 bit system har nok adresse muligheter i overskuelig fremtid.»<sup>5</sup>

## Videre lesning

<http://en.wikipedia.org/wiki/32-bit>

<http://en.wikipedia.org/wiki/64-bit>

<http://computer.howstuffworks.com/bytes.htm>

<http://computer.howstuffworks.com/bytes4.htm> (lenker til enda mer informasjon)

---

1 Kilde: <http://no.wikipedia.org/wiki/Bit>

2 «In the **ASCII character set**, each binary value between 0 and 127 is given a specific character. Most computers extend the ASCII character set to use the full range of 256 characters available in a byte. The upper 128 characters handle special things like accented characters from common foreign languages.» Kilde: <http://computer.howstuffworks.com/bytes2.htm>

3 Prøv å skrive en tekstfil, lagret som txt, og skriv en kort tekst på 29 tegn. Lagre den. Den skal så få en størrelse på 30 byte. Legg til ett ekstra tegn og det blir 31 byte, en byte til for neste tegn.

4 Helt nøyaktig tilsvarer det: 4 294 967 295.

5 Kilde: <http://www.klikk.no/teknologi/data/article550145.ece>

## Nøyaktige datastørrelser<sup>6</sup>

Benevnelse	Størrelse i tall	Tierplasser
Kilo (K)	$2^{10} = 1\ 024$	4
Mega (M)	$2^{20} = 1\ 048\ 576$	7
Giga (G)	$2^{30} = 1\ 073\ 741\ 824$	10
Tera (T)	$2^{40} = 1\ 099\ 511\ 627\ 776$	13
Peta (P)	$2^{50} = 1\ 125\ 899\ 906\ 842\ 624$	16
Exa (E)	$2^{60} = 1\ 152\ 921\ 504\ 606\ 846\ 976$	19
Zetta (Z)	$2^{70} = 1\ 180\ 591\ 620\ 717\ 411\ 303\ 424$	22
Yotta (Y)	$2^{80} = 1\ 208\ 925\ 819\ 614\ 629\ 174\ 706\ 176$	25

Et 16-bits tall kan lagre  $2^{16}$ , altså 65,536 ulike tall.

---

6 Kilde: <http://computer.howstuffworks.com/bytes3.htm>