(Positions) Talsystemer

For IT studerende

Henrik Kressner
Denne og andre kan findes på: https://synkro.dk/bog

Indholdsfortegnelse

Indledning

Selv om de færreste tænker over det i hverdagen, går vi og arbejder med forskellige talsystemer. Det mest kendte er nok titaltsystemet, men vi bruger til daglig både 7, 12, 24, 60 og 365 talsystemet, i forbindelse med begrebet tid. En uge er syv dage, en dag er 2 gange tolv eller 24 timer, der er 60 minutter på en time, 60 sekunder på et minut, og året har (for det meste) 365 dage.

Ovenstående bruges ikke bare i dagligt regnearbejde inden for økonomi, men også inden for astronomi og navigation benyttes de systemer rutinemæssigt.

Computerne er nok mest kendt for at bruge 2 tal systemet, men både 8 og 16 tal systemet er meget brugt i forbindelse med computere.

Det første kapitel gennemgår teorien for positions talsystemer, ved hjælp af 10 talsystemet. Derefter følger korte kapitler med eksempler på 2, 8 og 16 talsystemet, fordi det er dem vi bruger i forbindelse med computere.

Dette kompendie kan findes på: https://synkro.dk/bog og kan frit bruges af enhver der har løst til det.

Som altid er forslag og kritik velkommen.

Henrik Kressner
kressner@synkro.dk
Morud sommeren 2016

Positions talsystem - Generelt

I et positionstalsystem har ikke bare tallet (symbolet) selv betydning, dets placering har også betydning.

I titalsystemt. også kaldet decimalsystemet, har vi symbolerne 0 (nul), 1 (et), 2 (to), 3 (tre), 4 (fire), 5 (fem), 6 (seks), 7 (syv), 8 (otte) og 9 (ni). Vi har altså brug for lige så mange symboler som grundtallet.

Vi kan kombinere tallene og eksempelvis skrive 19, vi ved straks der menes nitten, men det er faktisk ikke det der står.

Der står 9 plus 1 gange 10 hvilket jo summer op til nitten.

Vi kan skrive det lidt mere matematisk korrekt, så kommer der til at stå:

$$9*10^0 + 1 * 10^1$$

Eller vi kan skrive det på denne måde:

$9 * 10^0$	=	9
$+ 1 * 10^1$	=	10

Sum	=	19

Vi kan se 10 tallet går igen i en stigende potens, alt efter hvor langt til venstre vi kommer, derfor kaldes tallet 10 for grundtallet, grundtallet giver navnet til talsystemet.

Hvis vi tager et større tal eksempelvis 2648 ser det således ud:

$8 * 10^0$	=	8
$+ 4 * 10^1$	=	+40
$+ 6 * 10^2$	=	+600
$+ 2 * 10^3$	=	+2000
---		---
2648		2648

For decimalsystemet gælder generelt:

Hvis vi vil gange med ti flytte vi alle symboler et felt til venstre, og sætter et nul på den ledige plads.

Hvis vi vil dividere med 10, flytter vi alle symboler en plads til højre.

Hvis vi kun arbejder med heltals division, smider vi det ciffer væk der stod helt til højre, arbejder vi med decimaler sætter vi et komma.

Generelt for et positionstalsystem gælder:

Hvis vi kalder symbolet i et talsystem bogstavet S, grundtallet G, og posistions-nummeret n, kan det generelt skrives på denne form:

$$S_n * G^n + S_{(n-1)} * G^{(n-1)} + ... + S_0 * G^0$$

Eller matematisk lidt smukkere:

$$\sum_{p=0}^{p=n} (S_p * G^p)$$

Hvis ovenstående er indlysende er der ingen grund til at læse mere, men det er tit rart med nogle eksempler, de kommer i de næste kapitler.

2 talsystemet / Det binære talsystem

Digitale computere bygger på 2 totalsystemet, det vil sige noget enten kan være nul eller et, hvis der opstår andre situationer er det en fejl.

Grundtallet er altså 2 og de to symboler vi bruger er normalt 0 (nul) og 1 (et) men, der er en del andre måder at repræsentere totalsystemet på.

0	False	Low	Down	Open	Ja	Off
1	True	High	Up	Closed	Nej	On

Det kan opleves man bruger en subscript for at vise hvilket talsystem der arbejdes med, det undlades ofte i praksis i håb om omgivelserne antyder talsystemet

Eksempel: 11011_2

		Binært
$1 * 2^0$	=	1
$+ 1 * 2^1$	=	+10
$+ 0 * 2^2$	=	+0
$+ 1 * 2^3$	=	+1000
$+ 1 * 2^4$	=	+10000
---		---
11011		11011

Ofte ser man et ciffer i totalsystemet bliver kaldt en bit.

For det binære talsystem gælder generelt:

Hvis vi vil gange med 2 flytter vi alle symboler et felt til venstre, og sætter et nul på den ledige plads.

Hvis vi vi dividere med 2, flytter vi alle symboler en plads til højre, og smider det ciffer der stod længst til højre væk. Vi kan ikke arbejde med kommatal i totalsystemet.

8 talsystemet / Det octale

I ottetalsystemt har vi symbolerne 0 (nul), 1 (et), 2 (to), 3 (tre), 4 (fire), 5 (fem), 6 (seks) og 7 (syv), det betyder at 67_8 **ikke** betyder syvogtres.

8 talsystemt er en udvikling af det binære talsystem, det består af tre bit, som hver kan have værdien 0 eller 1.

Med 3 bit har man 8 kombinationsmuligheder i det $2*2*2 = 2^3 = 8$. Heraf grundtallet.

Eksempel: 3267_8

		Octalt
$7 * 8^0$	=	7
$+ 6 * 8^1$	=	+60
$+ 2 * 8^2$	=	+200
$+ 3 * 8^3$	=	+3000
---		---
3267		3267

Bemærk: Når der er sat et mellerum for hver 3 digit i det binære tal skyldes det, at der går tre binære cifre til et octal tal, ved at lave et mellemrum for hver 3 digits, giver det mere overblik.

For det octale talsystem gælder generelt:

Hvis vi vil gange med 8 flytter vi alle symboler et felt til venstre, og sætter et nul på den ledige plads.

Hvis vi vi dividere med 8, flytter vi alle symboler en plads til højre, og smider det ciffer der stod længst til højre væk. Vi kan ikke arbejde med kommatal.

16 talsystemet / Det hexadecimale

For de fleste computerteknikere er det hexadecimale system nok det mest udbredte, i daglig tale kaldes det blot hex.

Det hexadecimale system bygger som det octale system på det binære system, hvor vi i det octale system havde 3 bit per grundtal, er der 4 bit i det hexadecimale system, hvilket medfører vi har $2^4 = 16$ kombinationsmuligheder.

Det betyder vi må finde på nye symboler, det er gjort ved at benytte decimaltal-symboler fra nul til og med 9, 10 betegnes med bogstavet A, 11 med B o.s.v frem til 15 der betegnes med bogstavet F.

Ofte ser man et hex tal afsluttes med et H for at vise der er tale om et hexadecimalt tal.

Eksempel: $E37D_{16}$ = E37DH

		Hexadecimal
$D * 16^0$	=	15
$+ 7 * 16^1$	=	+160
$+ 3 * 16^2$	=	+1.400
$+ E * 16^3$	=	+160 000
---		---
E37DH	=	161.575

For det hexadecimale talsystem gælder generelt:

Hvis vi vil gange med 16 flytte vi alle symboler et felt til venstre, og sætter et nul på den ledige plads.

Hvis vi vi dividere med 16, flytter vi alle symboler en plads til højre, og smider det ciffer der stod længst til højre væk. Vi kan ikke arbejde med kommatal.

De første 16 tal i de 4 talsystemer ser således ud:

Binær	Octal	Decimal	Hexadecimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

Omregning mellem talsystemer

Det er altid nemt at omregne fra et vilkårligt talsystem til titalsystemet. Det er straks lidt mere drilsk når man vil fra titalsystemet til et af de andre systemer, selv om det faktisk bare er den omvendte operation.

Eksempel 1: Vi vil omdanne 237 fra decimaltal til binært:

Først skal vi finde den højeste potens af to der går op i 237, vi laver en lille tabel:

2^0	=	1
2^1	=	2
2^2	=	4
2^3	=	8
2^4	=	16
2^5	=	32
2^6	=	64
2^7	=	128
2^8	=	256

Af tabellen kan det ses at 237 svarer til 2^7 plus et eller andet, vi ved altså det binære tal vi leder efter må se således ud:

$$2^7 = 128 = 1000\ 0000 + \text{etellerandet.}$$

Vi skal ikke være den store matematiker for at kunne konstatere, at etellerandet må være = 237 minus 128 = 109

Vi kan straks se 64 < 109 < 128, så næste binære tal må være 64 = $100\ 0000_2$

Vi kan også se der stadig er en rest, nemlig 109 - 64 = 45.

32 < 45 < 64, så vi har nu $10\ 0000_2$

Der er stadig en rest nemlig 45 - 32 = 13

Vi finder det sidste tal: 8 < 13 < 16, så vi har nu 1000_2

resten er 13 - 8 = 5 = 101_2

Vi tager nu de binære tal vi har fundet og lægger dem sammen:

	1000 0000	=	2^7	=	128
+	100 0000	=	2^6	=	64
+	10 0000	=	2^5	=	32
+	1000	=	2^3	=	8
+	101	=			5
	---			---	
	1110 1101	=			237

Og så har vi konverteret 237 fra titalsystemet til 1110 1101 i totalsystemt.

Arbejdet med at konvertere er ret træls at lave i hånden, men en computer kan nemt programmeres til at udføre beregningen.

Eksempel 2:

Vi vil omdanne 237 fra decimaltal til Hex:

Først skal vi finde den højeste potens af 16 der går op i 237, vi laver en lille tabel:

	Symbol	Decimalt
16^0	= 1	1
16^1	= F1	16
16^2	= FF1	256

Da talsystemet starter med nul trækker vi en fra antallet af kombinationsmuligheder:

256 - 1 = 255 = FFH

Vi kan også være ”frække” og gøre det på en anden måde.

255 - 237 = 18

der skal altså trækkes 18 = 13H fra 255 = FFH

	FFH	1111 1111
-	12H	0001 0011
	---	---
	EDH	1110 1100

Opgaver

1. Undersøg om den app du har til at regne med, kan konvertere mellem talsystemer. Hvis den ikke kan, find en der kan.

2. Test din app med eksemplerne i dette kompendie.