Introduktion til Programmering og Problemløsning (PoP)

Jon Sporring
Department of Computer Science
2021/09/16

UNIVERSITY OF COPENHAGEN

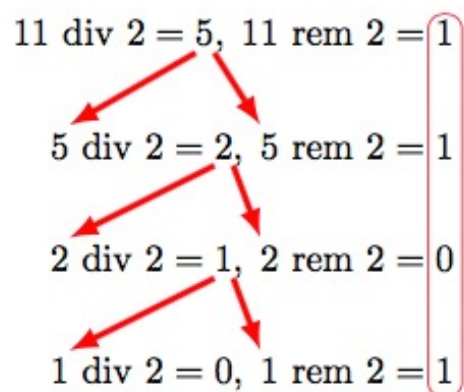# Hvor langt er I kommet med materialet?

https://tinyurl.com/ybjpr8m9

Alt på computeren er relateret til binære tal

https://tinyurl.com/ycpxcto5

Antal tal ved n bits: 2^n

Dec	Bin	Oct	Hex	Dec	Bin	Oct	Hex
0	0	0	0	32	100000	40	20
1	1	1	1	33	100001	41	21
2	10	2	2	34	100010	42	22
3	11	3	3	35	100011	43	23
4	100	4	4	36	100100	44	24
5	101	5	5	37	100101	45	25
6	110	6	6	38	100110	46	26
7	111	7	7	39	100111	47	27
8	1000	10	8	40	101000	50	28
9	1001	11	9	41	101001	51	29
10	1010	12	a	42	101010	52	2a
11	1011	13	b	43	101011	53	2b
12	1100	14	c	44	101100	54	2c
13	1101	15	d	45	101101	55	2d
14	1110	16	e	46	101110	56	2e
15	1111	17	f	47	101111	57	2f
16	10000	20	10	48	110000	60	30
17	10001	21	11	49	110001	61	31
18	10010	22	12	50	110010	62	32
19	10011	23	13	51	110011	63	33
20	10100	24	14	52	110100	64	34
21	10101	25	15	53	110101	65	35
22	10110	26	16	54	110110	66	36
23	10111	27	17	55	110111	67	37
24	11000	30	18	56	111000	70	38
25	11001	31	19	57	111001	71	39
26	11010	32	1a	58	111010	72	3a
27	11011	33	1b	59	111011	73	3b
28	11100	34	1c	60	111100	74	3c
29	11101	35	1d	61	111101	75	3d
30	11110	36	1e	62	111110	76	3e
31	11111	37	1f	63	111111	77	3f

Dividér med 2 algoritmen

$$11 \text{ div } 2 = 5, \quad 11 \text{ rem } 2 = 1$$
$$5 \text{ div } 2 = 2, \quad 5 \text{ rem } 2 = 1$$
$$2 \text{ div } 2 = 1, \quad 2 \text{ rem } 2 = 0$$
$$1 \text{ div } 2 = 0, \quad 1 \text{ rem } 2 = 1$$

https://tinyurl.com/y7s5979a

Hvad gøre dividér med 2?

Eksempler:

$4 = 0100_2$
$4/2 = 2 = 0010_2$

$12 = 1100_2$
$12/2 = 6 = 0110_2$

$3 = 0011_2$
$3/2 = 1.5 = 0001.1_2$

Heltal og endelig præcision

Bytes er 8 bit => 2^8 forskellige tal

0b01111111uy;;
0b01111111y;;
0b10000000uy;;
0b10000000y;;
0b10000001uy;;
0b10000001y;;

let a = 50uy

let b = a+110uy

let c = b + 110uy;;

let a = 50y

let b = a+110y

let c = b + 110y;;

Type	syntax	Examples	Value
int, int32	`<int	hex>`	3, 0x3	3
	`<int	hex>l`	3l, 0x3l	
uint32	`<int	hex>u`	3u	3
	`<int	hex>ul`	3ul	
byte, uint8	`<int	hex>uy`	97uy	97
	`'<char>'B`	'a'B	
byte[]	`"<string>"B`	"a\n"B	[|97uy; 10uy|]
	`@"<string>"B`	@"a\n"B	[|97uy; 92uy; 110uy|]
sbyte, int8	`<int	hex>y`	3y	3
int16	`<int	hex>s`	3s	3
uint16	`<int	hex>us`	3us	3
int64	`<int	hex>L`	3L	3
uint64	`<int	hex>UL`	3UL	3
	`<int	hex>uL`	3uL	
float, double	`<float>`	3.0	3.0
	`<hex>LF`	0x013fLF	9.387247271e-323
single, float32	`<float>F`	3.0F	3.0
	`<float>f`	3.0f	3.0
	`<hex>lf`	0x013flf	4.4701421e-43f
decimal	`<float	int>M`	3.0M,3M	3.0
	`<float	int>m`	3.0m,3m	
string	`"<string>"`	"\"quote\".\n"	"quote".<newline>
	`@"<string>"`	@"""quote"".\n"	"quote".\n.
	`"""<string>"""`	""""quote".\n"""	"quote".\n

Floats og problemet med endelig præcision

Floats er 64 bit => 2^{64} forskellige tal

```
let a = 1.0
let b = 1e-10
let c = a+b;;
printfn "a=%A, b=%A , c=%A " a b c;;

printfn "%A" (c-1.0);;


let a = 1e10
let b = a/1e-5
let c = b - 1e15;;
```

```
Konsekvens:
  let a = 3.0+1e-15;;
  a = 3.0;;
  abs (a - 3.0) < 1e-10;;
```

Resumé

Vi har talt om tal:

- Hvor mange man kan have på n bits

- Repeteret divider-med-2 algoritmen

- Overflow i heltalsberegninger

- Regnefejl i floats og konsekvens for sammenligning af floats

Spørgetime

https://tinyurl.com/4sr3h6rm