

Introduktion til Programmering og Problemløsning (PoP)

Jon Sparring
Department of Computer Science
2020/09/04

UNIVERSITY OF COPENHAGEN



Alt på computeren er relateret til binære tal

Decimal:

- Base – 10
- Alphabet – {0,1,...,9}

$d_n d_{n-1} \dots d_0 . d_{-1} d_{-2} \dots d_{-m}$

117.42

$$v = \sum_{i=-m}^n d_i 10^i.$$

Binær:

- Base – 2
- Alphabet – {0,1}

$b_n b_{n-1} \dots b_0 . b_{-1} b_{-2} \dots b_{-m}$

101.11

$$v = \sum_{i=-m}^n b_i 2^i,$$

Et heksal ciffer

Et oktal ciffer

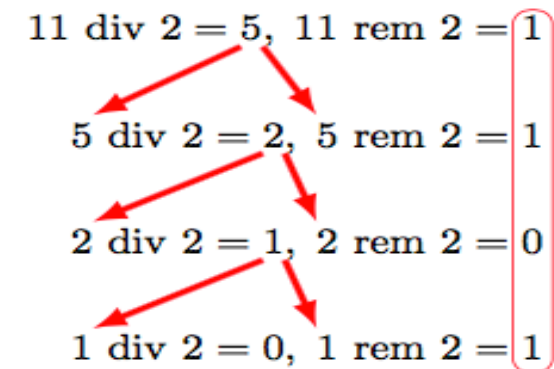
Dec	Bin	Oct	Hex	Dec	Bin	Oct	Hex
0	0	0	0	32	100000	40	20
1	1	1	1	33	100001	41	21
2	10	2	2	34	100010	42	22
3	11	3	3	35	100011	43	23
4	100	4	4	36	100100	44	24
5	101	5	5	37	100101	45	25
6	110	6	6	38	100110	46	26
7	111	7	7	39	100111	47	27
8	1000	10	8	40	101000	50	28
9	1001	11	9	41	101001	51	29
10	1010	12	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

Fra binær til decimal

$$v = \sum_{i=-m}^n b_i 2^i,$$

Fra decimal til binær



Octal og Hexadecimale tal

Octaltal:

Base: 8

Alphabet: {0,1,2,3,4,5,6,7}

Octalciffer svarer til 3 bit:

$10111_2 \rightarrow 010\ 111_2 \rightarrow 27_8$

Hexadecimale tal:

Base: 16

Alphabet: {0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f}

Octalciffer svarer til 4 bit:

$10111_2 \rightarrow 0001\ 0111_2 \rightarrow 17_{16}$

Typer definerer formen på klodsen

Type:	int	float	char	string	float	float
Værdi:	3	3.0	'3'	"3"	3e0	3.0e0

Metatype	Type name	Description
Boolean	<code>bool</code>	Boolean values true or false
Integer	<code>int</code>	Integer values from -2,147,483,648 to 2,147,483,647
	<code>byte</code>	Integer values from 0 to 255
	<code>sbyte</code>	Integer values from -128 to 127
	<code>int8</code>	Synonymous with <code>sbyte</code>
	<code>uint8</code>	Synonymous with <code>byte</code>
	<code>int16</code>	Integer values from -32768 to 32767
	<code>uint16</code>	Integer values from 0 to 65535
	<code>int32</code>	Synonymous with <code>int</code>
	<code>uint32</code>	Integer values from 0 to 4,294,967,295
Real	<code>int64</code>	Integer values from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
	<code>uint64</code>	Integer values from 0 to 18,446,744,073,709,551,615
	<code>float</code>	64-bit IEEE 754 floating point value from $-\infty$ to ∞
	<code>double</code>	Synonymous with <code>float</code>
	<code>single</code>	A 32-bit floating point type
Character	<code>float32</code>	Synonymous with <code>single</code>
	<code>decimal</code>	A floating point data type that has at least 28 significant digits
None	<code>char</code>	Unicode character
	<code>string</code>	Unicode sequence of characters
Object	<code>unit</code>	The value ()
Exception	<code>obj</code>	An object
	<code>exn</code>	An exception

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

ASCII, Latin1, UTF8 og Kodesider

Character	Escape sequence	Description
BS	\b	Backspace
LF	\n	Line feed
CR	\r	Carriage return
HT	\t	Horizontal tabulation
\	\\	Backslash
"	\"	Quotation mark
'	\'	Apostrophe
BEL	\a	Bell
FF	\f	Form feed
VT	\v	Vertical tabulation
	\uXXXX, \UXXXXXXXX, \DDD	Unicode character

x0+0x	00	10	20	30	40	50	60	70
00	NUL	DLE	SP	0	@	P	'	p
01	SOH	DC1	!	1	A	Q	a	q
02	STX	DC2	"	2	B	R	b	r
03	ETX	DC3	#	3	C	S	c	s
04	EOT	DC4	\$	4	D	T	d	t
05	ENQ	NAK	%	5	E	U	e	u
06	ACK	SYN	&	6	F	V	f	v
07	BEL	ETB	'	7	G	W	g	w
08	BS	CAN	(8	H	X	h	x
09	HT	EM)	9	I	Y	i	y
0A	LF	SUB	*	:	J	Z	j	z
0B	VT	ESC	+	;	K	[k	{
0C	FF	FS	,	<	L	\	l	
0D	CR	GS	-	=	M]	m	}
0E	SO	RS	.	>	N	^	n	~
0F	SI	US	/	?	O	_	o	DEL

x0+0x	80	90	A0	B0	C0	D0	E0	F0
00			NBSP	°	À	Đ	à	đ
01			¡	±	Á	Ñ	á	ñ
02			¢	²	Â	Ò	â	ò
03			£	³	Ã	Ó	ã	ó
04			¤	´	Ä	Ô	ä	ô
05			¥	µ	Å	Õ	å	õ
06				¶	Æ	Ö	æ	ö
07			§	·	Ç	×	ç	÷
08			¨	¸	È	Ø	è	ø
09			©	¹	É	Ù	é	ù
0a			ª	º	Ê	Ú	ê	ú
0b			«	»	Ë	Û	ë	û
0c			¬	$\frac{1}{4}$	Ì	Ü	ì	ü
0d			SHY	$\frac{1}{2}$	Í	Ý	í	ý
0e			®	$\frac{3}{4}$	Î	Þ	î	þ
0f			¯	¸	Ï	ß	ï	ÿ

Table C.1: ASCII

Operatorer og præcedens

Operatorer og typer

$3 + 4$

$3.0 + 4.0$

$3 + 4.0$

$5 / 2$

$5 \% 2$

$2 * (5 / 2) + 5 \% 2$

$2.0 ** 3.0$

`pown 2 3`

`"hej " + "med " + "dig"`

Operatorer og præcedens

Operatorer og typer	Præcedens og association
3 + 4	exp 0.0
3.0 + 4.0	exp 1.0
3 + 4.0	<u>(exp 0.0) + 1.0</u>
5 / 2	<u>2.0 ** (3.0 ** 4.0)</u>
5 % 2	<u>(2.0 / 3.0) / 4.0</u>
2 * (5 / 2) + 5 % 2	
2.0 ** 3.0	
pown 2 3	
"hej " + "med " + "dig"	

Operator	Associativity	Description
+<expr>, -<expr>, ~~~<expr>	Left	Unary identity, negation, and bitwise negation operator
f <expr>	Left	Function application
<expr> ** <expr>	Right	Exponent
<expr> * <expr>, <expr> / <expr>, <expr> % <expr>	Left	Multiplication, division and remainder
<expr> + <expr>, <expr> - <expr>	Left	Addition and subtraction binary operators
<expr> ^^^ <expr>	Right	bitwise exclusive or
<expr> < <expr>, <expr> <= <expr>, <expr> > <expr>, <expr> >= <expr>, <expr> = <expr>, <expr> <> <expr>, <expr> <<< <expr>, <expr> >>> <expr>, <expr> &&& <expr>, <expr> <expr> ,	Left	Comparison operators, bitwise shift, and bitwise 'and' and 'or'.
<expr> && <expr>	Left	Boolean and
<expr> <expr>	Left	Boolean or

String slicing, boolske værdier og operatorer

Slicing	Boolske værdier og operatorer	Sammenligninger
<code>"abcdefghijkl".[1]</code> = 'b'	<code>true = 1</code>	<code>3 < 4</code>
<code>"abcdefghijkl".[1..4]</code> = "bcde"	<code>false = 0</code>	<code>3 > 4</code>
<code>"abcdefghijkl".[..4]</code> = "abcde"	<code>a && b</code>	<code>3 <> 4</code>
<code>"abcdefghijkl".[4..]</code> = "efghijkl"	<code>a b</code>	<code>3 = 4</code>
<code>"abcdefghijkl".Length</code> = 12	<code>not a</code>	<code>not 3 = 4</code>
<code>"abcdefghijkl".[0..11]</code> = "abcdefghijkl"		<code>not (3 = 4)</code>

Resumé

I video har du hørt om:

- Talsystem (decimal, binære, octale, hexadecimal)
- En kort gennemgang af Absalonsiderne
- En gennemgang af den første arbejdsseddel
- En kort introduction til imperative programmering