Tegn og tekst

\yvind og]se N{rb} ?

Kapittel 2

27. August 2008

Posisjonssystemer

10^5 (100 000)	10^4 (10 000)	10^3 (1 000)	10^2 (100)	10^1 (10)	10^0 (1)

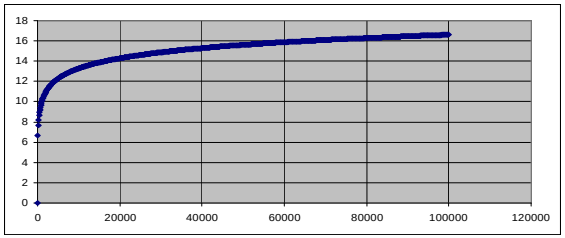
2^7 (128)	2^6 (64)	2^5 (32)	2^4 (16)	2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)

Bitposisjoner og bitmønstre

- Med n bitposisjoner, hvor mange ulike bitmønstre kan man lage?
- Hvis vi trenger k ulike bitmønstre (av lik lengde), hvor mange bitposisjoner trenger vi da?
- n sifre (biter/bitposisjoner) gir
  - tallene fra 0 til $2^n - 1$, dvs
  - 2^n ulike bitmønstre

Logaritmer – en kort repetisjon

- "Toerlogaritmen til k er det tallet du må opphøye 2 i for å få k."
- Funksjonen $n = \log_2(k)$:



Regning med binærtall
Addisjon

- Regneregler:
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 og 1 i mente
- Eksempler:

  5 + 2: 5 + 5: 7 + 7:

Regning med binærtall
Multiplikasjon

- Regneregler:
  - 0 * 0 = 0
  - 0 * 1 = 0
  - 1 * 0 = 0
  - 1 * 1 = 1
- Eksempler:

  5 * 5: 7 * 7:

Læringsmål (tegn og tekst)

- Forstå prinsippene for hvordan tegn og tekst kan representeres ved hjelp av bits og bytes.
- Kjenne til en del sentrale standarder, spesielt:
  - ASCII
  - ISO 8859 ”alfabetsuppen”
  - Unicode
- Kunne hovedtrekkene i hvordan disse standardene er bygget opp.

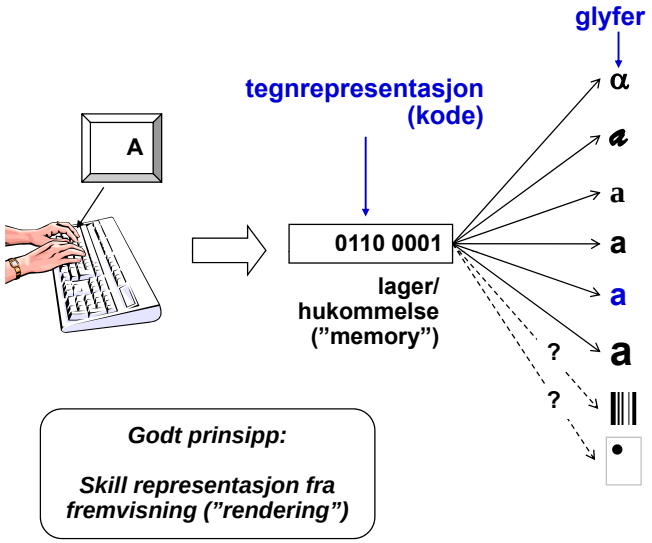
Problemstilling

- Utgangspunkt:
  Hvert tegn i teksten representeres av et unikt bitmønster.
- Eksempel:
  - E = 01000101
  - H = 01001000
  - I = 01001001
  - HEI = 01001000 01000101 01001001
- Sender (skriver) og mottaker (leser) må være enige om kodingen.
  - Vi trenger standarder!

Aktuelle spørsmål

- Hvilke tegn skal representeres?
- Hvor mange biter per tegn?
- Fast eller variabelt antall biter per tegn?
- Hvordan håndtere
  - tegn som er varianter av andre tegn?
  - ligaturer (sammensetninger)?
- Bør det være noen form for systematikk i bitmønstrene?

Om tegn og glyfer



Godt prinsipp:

Skill representasjon fra fremvisning ("rendering")

ASCII kodetabell

- American Standard Code for Information Interchange
- 1963 →

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

Linjeskift: LF og CR

- LF (Line Feed):
  - 0x0A
  - "Indicates movement of the printing mechanism or display cursor to the next line."
- CR (Carriage Return):
  - 0x0D
  - "Indicates movement of the printing mechanism or display cursor to the starting position of the same line."
- Linjeskift representeres i dag på ulike måter:
  - PC: CR + LF
  - Mac: CR
  - UNIX: LF

ISO 646-60 kodetabell

- Bygger på ASCII.
- [\ ] { | } er ofret til fordel for Æ Ø Å æ ø å
- Lignende tilpasninger er gjort i tilsvarende standarder for andre språkmiljøer.

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

ISO 8859-1 (Latin-1) kodetabell

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL	DLE	space	0	@	P	`	p			no break space	°	À	Ð	à	ð
1	SOH	DC1	!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2	STX	DC2	”	2	B	R	b	r			¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	un-		¦	¶	Æ	Ö	æ	ö
7	BEL	ETB	'	7	G	W	g	w	defined		§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x			¨	¸	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y			©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{			«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|		¬	¼	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}			-	½	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~			®	¾	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL			¯	¿	Ï	ß	ï	ÿ

ISO 8859-15 (Latin-9) kodetabell

ISO 8859-1 modernisert

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL	DLE	space	0	@	P	`	p			no break space	°	À	Ð	à	ð
1	SOH	DC1	!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2	STX	DC2	”	2	B	R	b	r			¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t			€	Ž	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	un-		Š	¶	Æ	Ö	æ	ö
7	BEL	ETB	'	7	G	W	g	w	defined		§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x			š	ž	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y			©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{			«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|		¬	Œ	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}			-	œ	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~			®	Ÿ	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL			¯	¿	Ï	ß	ï	ÿ

8 biter: Extended ASCII og ISO 8859

- 8859-1: Latin Alphabet No. 1 (Vest-Europa)
- 8859-2: Latin Alphabet No. 2 (slavisk, ungarsk, romansk)
- 8859-3: Latin Alphabet No. 3 (esperanto, maltesisk)
- 8859-5: Latin/Cyrillic Alphabet
- 8859-6: Latin/Arabic Alphabet
- 8859-7: Latin/Modern Greek Alphabet
- 8859-8: Latin/Hebrew Alphabet
- 8859-9: Latin Alphabet No. 5 (moderne tyrkisk)
- 8859-13: Latin Alphabet No. 7 (islandsk, grønlandsk, baltisk, nordsamisk)
- 8859-14: Latin Alphabet No. 8 (keltisk)
- 8859-15: Latin Alphabet No. 9 (modernisert 8859-1, med bl.a. euro-tegn)

ETSI GSM 03.38 kodetabell for SMS

	00	10	20	30	40	50	60	70
0	@	Δ	space	0	¡	P	¿	p
1	£	_	!	1	A	Q	a	q
2	$	Φ	"	2	B	R	b	r
3	¥	Γ	#	3	C	S	c	s
4	è	Λ	¤	4	D	T	d	t
5	é	Ω	%	5	E	U	e	u
6	ù	Π	&	6	F	V	f	v
7	ì	Ψ	'	7	G	W	g	w
8	ò	Σ	(	8	H	X	h	x
9	Ç	Θ	)	9	I	Y	i	y
A	LF	Ξ	*	:	J	Z	j	z
B	Ø	ESC	+	;	K	Ä	k	ä
C	ø	Æ	,	<	L	Ö	l	ö
D	CR	æ	-	=	M	Ñ	m	ñ
E	Å	undef	.	>	N	Ü	n	ü
F	å	É	/	?	O	§	o	à

…pluss disse 10 "escape"-sekvensene

€	ESC e
FF	ESC LF
[	ESC <
\	ESC /
]	ESC >
^	ESC Λ
{	ESC (
|	ESC @
}	ESC)
~	ESC =

Den endelige løsning? – Unicode og ISO 10646

- 21 biter, med mulighet for 1 114 112 kodepunkter
  - Ca 130 000 private
  - Ca 870 000 ennå ikke brukt
- Første 128 tegn er identisk med ASCII
- Første 256 tegn identisk med ISO 8859-1
- For hvert tegn finnes
  - en representativ glyf
  - kodepunktet
  - et navn
  - klassifisering
  - skriveretning

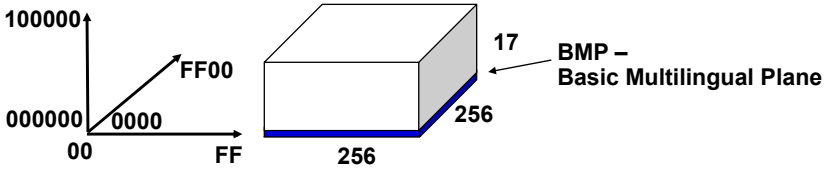
se http://www.unicode.org/charts/

- Vedtatte tegn med kodepunkter skal aldri endres

Unicode-kuben

- Tegnsettet er delt opp i 17 plan med 2^{16} = 65536 bitmønstre i hvert plan
- Plan 0: BMP – Basic Multilingual Plane U+0000 to U+FFFF
- Plan 1: SMP – Supplementary Multilingual Plane – historiske språk (f. eks. egyptiske hieroglyfer), musikk
- Plan 2: SIP – Supplementary Ideographic Plane – sjeldne kinesiske tegn
- Plan 14: SPP – Supplementary Special Purpose Plane – tag characters

I Unicode skriver vi U+ istedenfor 0x



Kodepunkt vs representasjon

- Kodepunkt:
  - Tegnets "numeriske" verdi.
  - Det som står i kodetabellen.
- Representasjon:
  - Hvordan tegnet representeres som et bitmønster.

Unicode Transformation Formats

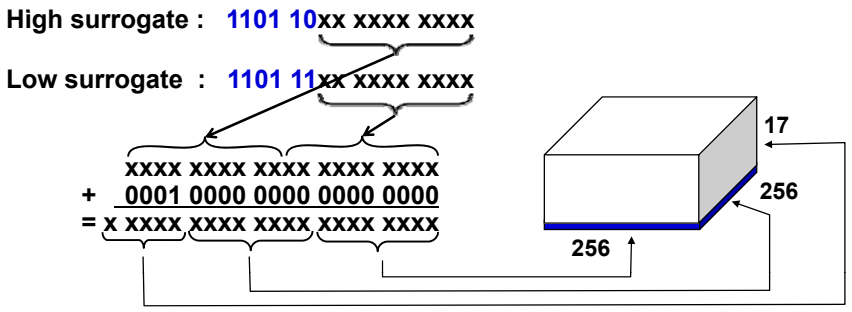
- UTF-32 (lite brukt):
  - Bruker 32 biter for alle tegn.
- UTF-16:
  - Tegn i BMP: 16 biter.
  - Tegn utenfor BMP: surrogatpar (32 biter).
- UTF-8:
  - Bruker 8, 16, 24, eller 32 biter avhengig av tegnet.
  - Tegnene i ASCII: 8 biter med ledende 0.

Unicode UTF-32

- Føy til ledende 0-biter opp til 32 biter foran representasjonen av kodepunktet
- Hva er representasjonen for A i UTF-32?
  (Kodepunktet for A er U+0041 = 0000 0000 0100 0001_2)
- Hva er representasjonen for 🏂 i UTF-32?
  (Kodepunktet for 🏂 er U+2658 = 0010 0110 0101 1000_2)

Unicode UTF-16

- Med 16 biter kan kodepunktene i BMP representeres direkte.
- De 16 planene over BMP adresseres med *surrogatpar*:
  - High surrogate – D800 – DBFF : 1101 10xx xxxx xxxx
  - Low surrogate – DC00 – DFFF: 1101 11xx xxxx xxxx
- Gir indirekte tilgang til 2^{20} = 1 048 576 ekstra tegn



Unicode UTF-16 – eksempel

- Hva er representasjonen for 𝄞 i Unicode UTF-16?
  (Kodepunktet for 𝄞 er U+1D11E = 1 1101 0001 0001 1110_2)

Unicode UTF-8

- Koding med variabel lengde
  - 8, 16, 24 eller 32 biter avhengig av kodepunktet

Unicode UTF-8 – eksempler

- Hva er representasjonen for Ø i Unicode UTF-8?
  (Kodepunktet for Ø er U+00D8 = 0000 0000 1101 1000_2)
- Hva er representasjonen for ♘ i Unicode UTF-8?
  (Kodepunktet for ♘ er U+2658 = 0010 0110 0101 1000_2)
- Hva er representasjonen for 𝄞 i Unicode UTF-16?
  (Kodepunktet for 𝄞 er U+1D11E = 1 1101 0001 0001 1110_2)

Big endian vs. Little endian

- I representasjoner som krever mer enn én byte, finnes det to mulige rekkefølger av bytene:
  - Starte med den mest signifikante (”Big endian”)
  - Starte med den minst signifikante (”Little/small endian”)
- Eksempel:
  UTF-16 Big endian for A er 0x 00 41
  UTF-16 Little endian for A er 0x 41 00
- Begge muligheter blir brukt i praksis, og dette kan gi problemer når data overføres fra et maskinmiljø til et annet!

Byte order mark (BOM)

- Et ”Byte order mark” (BOM) er tegnet ”Zero width no-break space” med kodepunkt U+FEFF i begynnelsen av en Unicode-fil.
- Siden det ikke finnes noe tegn med kodepunkt FFFE, kan BOM brukes til å finne filformatet (UTF-32, UTF-16, UTF-8 og Big eller Small endian):

Koding	BOM-bitmønster
UTF-32, big-endian	0x 00 00 FE FF
UTF-32, little-endian	0x FF FE 00 00
UTF-16, big-endian	0x FE FF
UTF-16, little-endian	0x FF FE
UTF-8	0x EF BB BF

Planen videre

- I dag har vi
  - gjort oss ferdige med binærtall og tallsystemer.
  - sett på representasjon av tegn og tekst.
- Les i læreboken om
  - Den historiske utviklingen frem til ASCII (kapittel 2.3-2.4)
  - Hva er et tegn? (kapittel 2.10.2-2.10.3)
- På plenumstimen torsdag/tirsdag får dere en introduksjon til å bestemme layout på nettsider ved hjelp av stilark (CSS).
- Neste onsdag ser vi på XML.