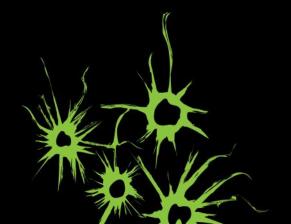


Binary-to-text encodings

Topic of Software Systems (TCS module 2)

Lecturer: Maarten Everts





REPRESENTING BINARY DATA (BYTE ARRAYS)

A byte-array:

byte[] data;

How to print its content?

Turning Strings into byte-arrays (using default encoding).

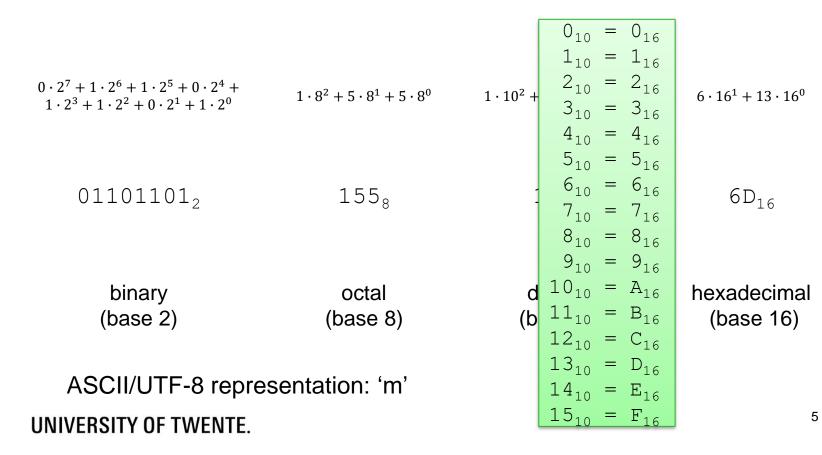
byte[] data = "Software Systemen".getBytes();

As a list of numbers?

23, 24, 37, 64, 62

le	rintabl		A		II control		
	acters	char			aracters	cha	
96	@	64	space	32	(Null character)	NULL	00
97	Α	65	!	33	(Start of Header)	SOH	01
98	В	66		34	(Start of Text)	STX	02
99	С	67	#	35	(End of Text)	ETX	03
100	D	68	\$	36	(End of Trans.)	EOT	04
101	E	69	%	37	(Enquiry)	ENQ	05
102	F	70	&	38	(Acknowledgement)	ACK	06
103	G	71	•	39	(Bell)	BEL	07
104	н	72	(40	(Backspace)	BS	80
105	1	73)	41	(Horizontal Tab)	HT	09
106	J	74	*	42	(Line feed)	LF	10
107	K	75	+	43	(Vertical Tab)	VT	11
108	L	76	,	44	(Form feed)	FF	12
109	M	77	-	45	(Carriage return)	CR	13
110	N	78		46	(Shift Out)	SO	14
111	0	79	1	47	(Shift In)	SI	15
112	P	80	0	48	(Data link escape)	DLE	16
113	Q	81	1	49	(Device control 1)	DC1	17
114	R	82	2	50	(Device control 2)	DC2	18
115	S	83	3	51	(Device control 3)	DC3	19
116	Т	84	4	52	(Device control 4)	DC4	20
117	U	85	5	53	(Negative acknowl.)	NAK	21
118	V	86	6	54	(Synchronous idle)	SYN	22
119	w	87	7	55	(End of trans. block)	ETB	23
120	Х	88	8	56	(Cancel)	CAN	24
121	Υ	89	9	57	(End of medium)	EM	25
122	Z	90	:	58	(Substitute)	SUB	26
123	[91	;	59	(Escape)	ESC	27
124	1	92	<	60	(File separator)	FS	28
125	1	93	=	61	(Group separator)	GS	29
126	۸	94	>	62	(Record separator)	RS	30
	_	95	?	63	(Unit separator)	US	31
					(Delete)	DEL	127

REPRESENTING NUMBERS (RECAP MODULE 1)



REPRESENTING BINARY DATA

A byte-array:

byte[] data;

How to print its content?

Hexadecimal!

Turning Strings into byte-arrays (using default encoding).

byte[] data = "Software Systemen".getBytes();

```
536f6674776172652053797374656d656e

1st byte (data[0]) 2nd byte (data[1])
```

	le	orintab		A		ASCII control					
		acters	cnar			characters (Null character)					
•	96	@	64	space	32	(Null character)	NULL	00			
a	97	Α	65	!	33	(Start of Header)	SOH	01			
b	98	В	66		34	(Start of Text)	STX	02			
C	99	С	67	#	35	(End of Text)	ETX	03			
d	100	D	68	\$	36	(End of Trans.)	EOT	04			
е	101	E	69	%	37	(Enquiry)	ENQ	05			
f	102	F	70	&	38	(Acknowledgement)	ACK	06			
g	103	G	71	•	39	(Bell)	BEL	07			
h	104	н	72	(40	(Backspace)	BS	80			
i	105	- 1	73)	41	(Horizontal Tab)	HT	09			
j	106	J	74	*	42	(Line feed)	LF	10			
k	107	K	75	+	43	(Vertical Tab)	VT	11			
- 1	108	L	76	,	44	(Form feed)	FF	12			
m	109	M	77	-	45	(Carriage return)	CR	13			
n	110	N	78		46	(Shift Out)	SO	14			
0	111	0	79	1	47	(Shift In)	SI	15			
р	112	P	80	0	48	(Data link escape)	DLE	16			
q	113	Q	81	1	49	(Device control 1)	DC1	17			
r	114	R	82	2	50	(Device control 2)	DC2	18			
s	115	S	83	3	51	(Device control 3)	DC3	19			
t	116	Т	84	4	52	(Device control 4)	DC4	20			
u	117	U	85	5	53	(Negative acknowl.)	NAK	21			
V	118	V	86	6	54	(Synchronous idle)	SYN	22			
w	119	W	87	7	55	(End of trans. block)	ETB	23			
X	120	X	88	8	56	(Cancel)	CAN	24			
У	121	Υ	89	9	57	(End of medium)	EM	25			
Z	122	Z	90	:	58	(Substitute)	SUB	26			
{	123	[91	;	59	(Escape)	ESC	27			
- 1	124	١.	92	<	60	(File separator)	FS	28			
}	125	1	93	=	61	(Group separator)	GS	29			
~	126	^	94	>	62	(Record separator)	RS	30			
		_	95	?	63	(Unit separator)	US	31			
						(Delete)	DEL	127			

	ASC	CII control	ASCII printable								
	cha	aracters			char	acters					
00	NULL	(Null character)	32	space	64	@	96	•			
01	SOH	(Start of Header)	33	!	65	Α	97	a			
02	STX	(Start of Text)	34		66	В	98	b			
03	ETX	(End of Text)	35	#	67	С	99	С			
04	EOT	(End of Trans.)	36	\$	68	D	100	d			
05	ENQ	(Enquiry)	37	%	69	E	101	е			
06	ACK	(Acknowledgement)	38	&	70	F	102	f			
07	BEL	(Bell)	39	•	71	G	103	g			
08	BS	(Backspace)	40	(72	H	104	h			
09	HT	(Horizontal Tab)	41)	73	1	105	i			
10	LF	(Line feed)	42	*	74	J	106	j			
11	VT	(Vertical Tab)	43	+	75	K	107	k			
12	FF	(Form feed)	44	,	76	L	108	- 1			
13	CR	(Carriage return)	45	-	77	M	109	m			
14	SO	(Shift Out)	46		78	N	110	n			
15	SI	(Shift In)	47	1	79	0	111	0			
16	DLE	(Data link escape)	48	0	80	P	112	р			
17	DC1	(Device control 1)	49	1	81	Q	113	q			
18	DC2	(Device control 2)	50	2	82	R	114	r			
19	DC3	(Device control 3)	51	3	83	S	115	s			
20	DC4	(Device control 4)	52	4	84	T	116	t			
21	NAK	(Negative acknowl.)	53	5	85	U	117	u			
22	SYN	(Synchronous idle)	54	6	86	V	118	v			
23	ETB	(End of trans. block)	55	7	87	W	119	w			
24	CAN	(Cancel)	56	8	88	X	120	x			
25	EM	(End of medium)	57	9	89	Υ	121	У			
26	SUB	(Substitute)	58	:	90	Z	122	Z			
27	ESC	(Escape)	59	;	91	[123	{			
28	FS	(File separator)	60	<	92	1	124	- 1			
29	GS	(Group separator)	61	=	93	1	125	}			
30	RS	(Record separator)	62	>	94	^	126	~			
31	US	(Unit separator)	63	?	95	_					
127	DEL	(Delete)									

Use more printable characters: base64

A MORE (SPACE) EFFICIENT BINARY-TO-TEXT ENCODING: BASE64

Value	Char	Value	Char	Value	Char	Value	Char
0	Α	16	Q	32	g	48	W
1	В	17	R	33	h	49	X
2	С	18	S	34	i	50	у
3	D	19	T	35	j	51	Z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	Н	23	X	39	n	55	3
8	I	24	Y	40	0	56	4
9	J	25	Z	41	р	57	5
10	K	26	а	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	С	44	S	60	8
13	N	29	d	45	t	61	9
14	0	30	е	46	u	62	+
15	Р	31	f	47	V	63	/

6 bits per character: $2^6 = 64$ different characters (26 + 26 + 10 + 2)

Text content	М						a								n							
ASCII	77 (0x4d))		97 (0x61)								110 (0x6e)							
Bit pattern	0 1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1 0
Index	19			22					5					46								
Base64-encoded	T			w					F						u							

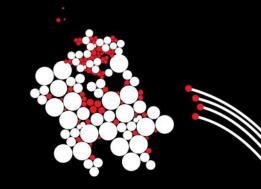
Note: padding is needed!

BASE64 EXAMPLES

NOTES ON ENCODING

Use libraries for this! (e.g., Apache Commons Codec library)

Superfluous remark: encoding is not encryption!



Binary-to-text encodings

Topic of Software Systems (TCS module 2)

Lecturer: Maarten Everts



