

Plan wykładu nr 4

- Systemy liczbowe
 - systemy pozycyjne i niepozycyjne (rzymski)
 - konwersje między systemami liczbowymi
- Jednostki informacji cyfrowej
 - bit, bajt słowo, FLOPS
- Kodowanie znaków
 - ASCII, ISO/IEC 646, ISO 8859
 - EBCDIC, Windows-1250, Unicode

Informatyka 1 (ES1E2009)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr II, studia stacjonarne I stopnia
Rok akademicki 2021/2022

Wykład nr 4 (12.05.2022)

dr inż. Jarosław Forenc

System dziesiętny (ang. decimal)

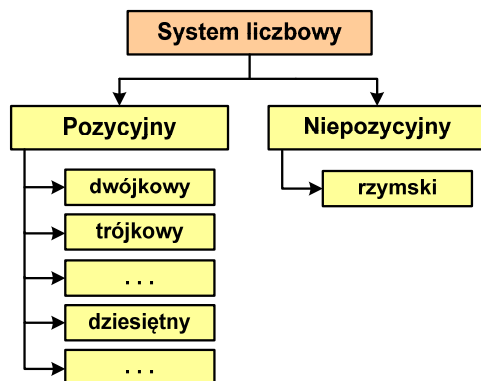
	10000	1000	100	10	1	0,1	0,01	0,001	0,0001	
	10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	← wagi
...	x_4	x_3	x_2	x_1	x_0	x_{-1}	x_{-2}	x_{-3}	x_{-4}	...
	4	3	2	1	0	-1	-2	-3	-4	← pozycje

- p - podstawa systemu pozycyjnego, D - zbiór dozwolonych cyfr
- $p = 10$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$$\begin{aligned}
 & \begin{array}{cccccc} 10^3 & 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} \\ \boxed{1} & \boxed{4} & \boxed{0} & \boxed{8} & \boxed{2} & \boxed{5} \end{array} \\
 1408,25_{(10)} &= \\
 &= \boxed{1 \cdot 10^3} + \boxed{4 \cdot 10^2} + \boxed{0 \cdot 10^1} + \boxed{8 \cdot 10^0} + \boxed{2 \cdot 10^{-1}} + \boxed{5 \cdot 10^{-2}} \\
 &= 1000 + 400 + 0 + 8 + 0,2 + 0,05
 \end{aligned}$$

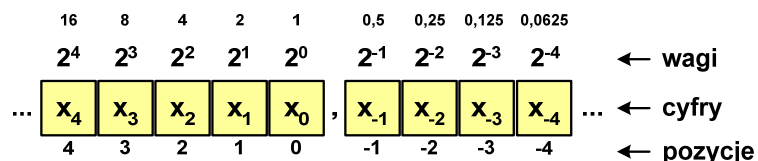
Systemy liczbowe

- **System liczbowy** - zbiór zasad umożliwiających zapis liczb za pomocą cyfr oraz wykonywanie działań na tych liczbach



- **Pozycyjny** - znaczenie cyfry jest zależne od miejsca (pozycji), które zajmuje ona w liczbie
 - system dziesiętny - liczba 111 (każda cyfra ma inne znaczenie)
- **Niepozycyjny** - znaczenie cyfry jest niezależne od miejsca położenia w liczbie
 - system rzymski - liczba III

System dwójkowy (ang. binary)

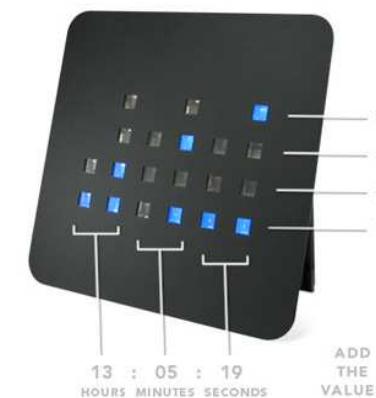


- w systemie dwójkowym: $p = 2$, $D = \{0, 1\}$

$$\begin{aligned}
 & \begin{array}{ccccccc} 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} \\ \hline 1 & 1 & 0 & 1 & , & 1 & 0 & 1 \end{array} \\
 1101,101_{(2)} = & \begin{array}{ccccccc} \boxed{1 \cdot 2^3} & + & \boxed{1 \cdot 2^2} & + & \boxed{0 \cdot 2^1} & + & \boxed{1 \cdot 2^0} & + & \boxed{1 \cdot 2^{-1}} & + & \boxed{0 \cdot 2^{-2}} & + & \boxed{1 \cdot 2^{-3}} \\ = & 8 & + & 4 & + & 0 & + & 1 & + & 0,5 & + & 0 & + & 0,125 \\ = & 13,625_{(10)} \end{array}
 \end{aligned}$$

System dwójkowy - zastosowania

- Powszechnie używany w informatyce, technice cyfrowej



System szesnastkowy (ang. hexadecimal)

- System heksadecymalny
- $p = 16$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Powszechnie używany w informatyce - jeden bajt można zapisać za pomocą tylko dwóch cyfr szesnastkowych

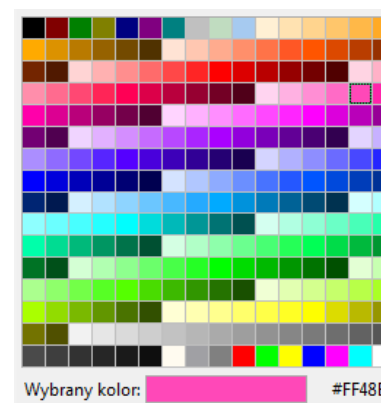
$$3A5D_{(16)} = 3 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16^1 + 13 \cdot 16^0 = 14941_{(10)}$$

- Sposoby zapisu liczb w systemie szesnastkowym:

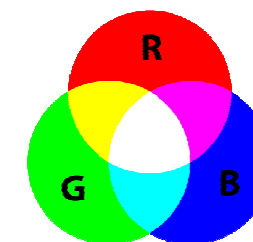
$3A5Dh$	$0x3A5D$	$\#3A5D$
$3A5D_{(16)}$	$3A5D_{16}$	$3A5D_{hex}$
$(3A5D)_{hex}$	$(3A5D)_{16}$	$\$3A5D$

System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Każda barwa przyjmuje wartość z zakresu: $0..255_{(10)}$, $00..FF_{(16)}$



#FF48B8



System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Kolory w dokumentach HTML:

```
<BODY bgcolor="#336699" text="#000000" link="#FFFF00"  
vlink="#33FFFF" alink="#FF0000">
```

ARCHIWUM

ENGLISH

Studia stacjonarne:

Poniedziałek:
12:15 - 14:00 [Informatyka 1 - wykład](#), sem. 2 ED, WE-Aula II

Wtorek:
08:30 - 10:00 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS3, WE-110
12:15 - 13:45 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS2, WE-110
14:15 - 15:45 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS1, WE-110

Przykład systemu niepozycyjnego - system rzymski

- W systemie rzymskim posługujemy się siedmioma znakami:
I - 1 V - 5 X - 10 L - 50 C - 100 D - 500 M - 1000
- Za pomocą dostępnych symboli można określić liczby od 1 do 3999
- System **addytywny** - wartość liczby określa się na podstawie sumy wartości cyfr, np.
 - II (1 + 1 = 2), XXX (10 + 10 + 10 = 30)
 - CLX (100 + 50 + 10 = 160), MMXII (1000 + 1000 + 10 + 1 + 1 = 2012)
- Wyjątkiem od powyższej zasady są liczby do opisu których używa się odejmowania, np.
 - IV (5 - 1 = 4), IX (10 - 1 = 9), XL (50 - 10 = 40), XC (100 - 10 = 90)
- Stosowany w łacińskiej części Europy do końca Średniowiecza
- Niewygodny w prowadzeniu nawet prostych działań arytmetycznych, brak ułamków

System szesnastkowy - zastosowania

- 48-bitowy adres fizyczny urządzenia (MAC - Media Access Control)

88:AD:D2:09:41:3B

producent numer egzemplarza

- <http://hwaddress.com>

HWAddress Home Countries Companies 88:AD:D2 Search

OUI	MAC range	Company
88-AD-D2	88-AD-D2-00-00-00 - 88-AD-D2-FF-FF-FF	Samsung Electronics Co.,Ltd

Przykład systemu niepozycyjnego - system rzymski

- Zasady tworzenia liczb:
 - zestawiamy odpowiednie znaki od oznaczającego liczbę największą do oznaczającego liczbę najmniejszą
 $XVI = 10(X) + 5(V) + 1(I) = 16$
 - jeżeli składnik liczby, którą piszemy, jest wielokrotnością liczby nominalnej, wtedy zapisywany jest z użyciem kilku następujących po sobie znaków
 $CCC = 100(C) + 100(C) + 100(C) = 300$
 - dodatkowo należy zachować zasadę nie pisania czterech tych samych znaków po sobie, lecz napisać jeden znak wraz ze znakiem oznaczającym wartość większą o jeden rząd liczbowy
 $CD = 500(D) - 100(C) = 400$

Przykład systemu niepozycyjnego - system rzymski

■ Zasady odczytu liczb:

- cyfry jednakowe są dodawane

$$MMM = 1000(M) + 1000(M) + 1000(M) = 3000$$

- cyfry mniejsze stojące przed większymi są odejmowane od nich

$$CDXCIV = 500(D) - 100(C) + 100(C) - 10(X) + 5(V) - 1(I) = 494$$

- cyfry mniejsze stojące za większymi są do nich dodawane

$$MDCLX = 1000(M) + 500(D) + 100(C) + 50(L) + 10(X) = 1660$$

Konwersja na system dziesiętny

- $p = 4, D = \{0, 1, 2, 3\}$

$$\begin{matrix} 4^4 & 4^3 & 4^2 & 4^1 & 4^0 \\ 2 & 1 & 3 & 0 & 2 \end{matrix}$$

$$21302_{(4)} = ?_{(10)}$$

$$21302_{(4)} = 2 \cdot 4^4 + 1 \cdot 4^3 + 3 \cdot 4^2 + 0 \cdot 4^1 + 2 \cdot 4^0$$

$$21302_{(4)} = 2 \cdot 16 + 0 \cdot 64 + 3 \cdot 16 + 1 \cdot 4 + 2 \cdot 1 = 626$$

$$21302_{(4)} = 2 + 0 + 48 + 64 + 512 = 626_{(10)}$$

- $p = 17, D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G\}$

$$\begin{matrix} 17^3 & 17^2 & 17^1 & 17^0 \\ A & C & 2 & 4 \end{matrix}$$

$$AC24_{(17)} = ?_{(10)}$$

$$AC24_{(17)} = 4 \cdot 17^3 + 2 \cdot 17^2 + 12 \cdot 17^1 + 10 \cdot 17^0$$

$$AC24_{(17)} = 4 \cdot 4913 + 2 \cdot 289 + 12 \cdot 17 + 10 \cdot 1 = 52636$$

$$AC24_{(17)} = 4 + 34 + 3468 + 49130 = 52636_{(10)}$$

Konwersja na system dziesiętny (schemat Hornera)

- $p = 4, D = \{0, 1, 2, 3\}$

$$21302_{(4)} = w_{(10)} \quad x_4 x_3 x_2 x_1 x_0 = w_{(10)}$$

$$w_{(10)} = 0$$

$$w_{(10)} = x_4 + w_{(10)} \cdot p = 2 + 0 \cdot 4 = 2$$

$$w_{(10)} = x_3 + w_{(10)} \cdot p = 1 + 2 \cdot 4 = 9$$

$$w_{(10)} = x_2 + w_{(10)} \cdot p = 3 + 9 \cdot 4 = 39$$

$$w_{(10)} = x_1 + w_{(10)} \cdot p = 0 + 39 \cdot 4 = 156$$

$$w_{(10)} = x_0 + w_{(10)} \cdot p = 2 + 156 \cdot 4 = 626_{(10)}$$

Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu $p = 10$ na system $p = 2$

$$626_{(10)} = ?_{(2)}$$

$$626_{(10)} = 1001110010_{(2)}$$

$$626 / 2 = 313 \text{ reszta } 0$$

$$313 / 2 = 156 \text{ reszta } 1$$

$$156 / 2 = 78 \text{ reszta } 0$$

$$78 / 2 = 39 \text{ reszta } 0$$

$$39 / 2 = 19 \text{ reszta } 1$$

$$19 / 2 = 9 \text{ reszta } 1$$

$$9 / 2 = 4 \text{ reszta } 1$$

$$4 / 2 = 2 \text{ reszta } 0$$

$$2 / 2 = 1 \text{ reszta } 0$$

$$1 / 2 = 0 \text{ reszta } 1$$

kolejność odczytywania
cyfr liczby w systemie
dwójkowym

kończymy, gdy liczba dziesiętna ma wartość 0

Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu $p = 10$ na system $p = 7$

$$626_{(10)} = ?_{(7)} \qquad 626_{(10)} = 1553_{(7)}$$

$626/7 = 89$	$\text{reszta } 3$	↑
$89/7 = 12$	$\text{reszta } 5$	
$12/7 = 1$	$\text{reszta } 5$	
$1/7 = 0$	$\text{reszta } 1$	

- zamiana liczby z systemu $p = 10$ na system $p = 14$

$$626_{(10)} = ?_{(14)} \qquad 626_{(10)} = 32A_{(14)}$$

$626/14 = 44$	$\text{reszta } 10 \rightarrow A$	↑
$44/14 = 3$	$\text{reszta } 2$	
$3/14 = 0$	$\text{reszta } 3$	

Jednostki informacji - bit

- Bit** (ang. binary digit) - podstawowa jednostka informacji stosowana w informatyce i telekomunikacji
- Określa najmniejszą ilość informacji potrzebną do stwierdzenia, który z dwóch możliwych stanów przyjął układ
- Bit przyjmuje jedną z dwóch wartości:
 - 0 (zero)
 - 1 (jeden)
- Bit jest tożsamy z cyfrą w systemie dwójkowym
- Oznaczenia bitów:
 - standard IEEE 1541 (2002) - mała litera „b”
 - standard IEC 60027 - „bit”

Szybkie konwersje: $2 \rightarrow 4, 8, 16$ $4, 8, 16 \rightarrow 2$

$2 \rightarrow 4$

$$110110011_{(2)} = ?_{(4)}$$

01	10	11	00	11
1	2	3	0	3

$$110110011_{(2)} = 12303_{(4)}$$

$2 \rightarrow 8$

$$10110011_{(2)} = ?_{(8)}$$

010	110	011
2	6	3

$$10110011_{(2)} = 263_{(8)}$$

$2 \rightarrow 16$

$$1011010_{(2)} = ?_{(16)}$$

0101	1010
5	A

$$1011010_{(2)} = 5A_{(16)}$$

$4 \rightarrow 2$

$$12303_{(4)} = ?_{(2)}$$

01	10	11	00	11
1	2	3	0	3

$$12303_{(4)} = 110110011_{(2)}$$

$8 \rightarrow 2$

$$263_{(8)} = ?_{(2)}$$

010	110	011
2	6	3

$$263_{(8)} = 10110011_{(2)}$$

$16 \rightarrow 2$

$$5A_{(16)} = ?_{(2)}$$

0101	1010
5	A

$$5A_{(16)} = 1011010_{(2)}$$

Jednostki informacji - bit

- Wielokrotności bitów:

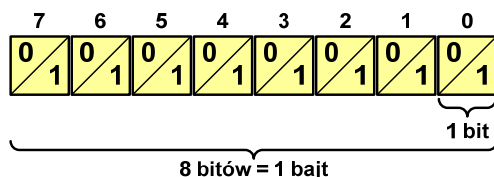
Przedrostki dziesiętne (układ SI)		
Nazwa	Symbol	Mnożnik
bit	b	---
kilobit	kb	$10^3 = 1000^1$
megabit	Mb	$10^6 = 1000^2$
gigabit	Gb	$10^9 = 1000^3$
terabit	Tb	$10^{12} = 1000^4$
petabit	Pb	$10^{15} = 1000^5$
eksabit	Eb	$10^{18} = 1000^6$
zettabit	Zb	$10^{21} = 1000^7$
jottabit	Yb	$10^{24} = 1000^8$

Przedrostki binarne (IEC 60027-2)		
Nazwa	Symbol	Mnożnik
bit	b	---
kibibit	Kib	$2^{10} = 1024^1$
mebibit	Mib	$2^{20} = 1024^2$
gibibit	Gib	$2^{30} = 1024^3$
tebibit	Tib	$2^{40} = 1024^4$
pebibit	Pib	$2^{50} = 1024^5$
eksbibit	Eib	$2^{60} = 1024^6$
zebibit	Zib	$2^{70} = 1024^7$
jobibit	Yib	$2^{80} = 1024^8$

- Przedrostki binarne** - wprowadzone w 1999 roku w celu odróżnienia przedrostków o mnożniku 1000 (10^3) od przedrostków o mnożniku 1024 (2^{10})

Jednostki informacji - bajt

- **Bajt** (ang. byte) - najmniejsza adresowalna jednostka informacji pamięci komputerowej składająca się z bitów
- W praktyce przyjmuje się, że jeden bajt to 8 bitów

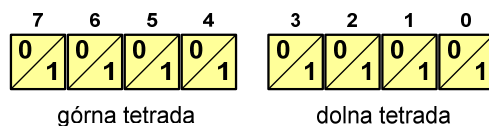


- Za pomocą jednego bajtu można zapisać $2^8 = 256$ różnych wartości:

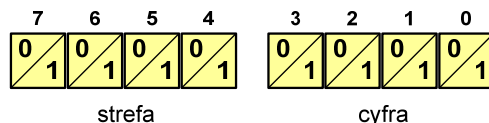
0000 0000	→	0
0000 0001	→	1	1111 1101	→ 253
0000 0010	→	2	1111 1110	→ 254
...	1111 1111	→ 255

Jednostki informacji - tetrada

- Bajt 8-bitowy można podzielić na dwie połówki 4-bitowe nazywane **tetradami** (ang. nibbles)
- Rozróżniamy bardziej znaczącą (górną) i mniej znaczącą (dolną) tetradę

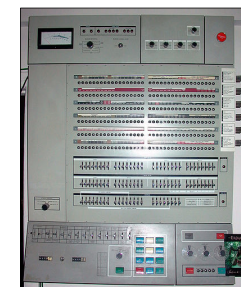


- Spotyka się też określenie **strefa** i **cyfra**



Jednostki informacji - bajt

- W pierwszych komputerach bajt mógł mieć inną liczbę bitów: 4, 6, 7, 9, 12
- 8-bitowy bajt:
 - koniec 1956 r. - pierwsze zastosowanie
 - 1964 r. - uznanie za standard (IBM System/360)
- Inna nazwa 8-bitowego bajtu - **oktet**
- Najczęściej stosowanym skrótem dla bajtu jest wielka litera „B”
 - „B” używane jest także do oznaczania **bela** - jednostki miary wielkości ilorazowych
 - zamiast belła częściej używa się jednostki podwielokrotnej - **decybel** (dB) więc nie ma problemu z rozróżnieniem obu jednostek



Jednostki informacji - bajt

- Wielokrotności bajtów:

Przedrostki dziesiętne (układ SI)		
Nazwa	Symbol	Mnożnik
bajt	B	---
kilobajt	kB	$10^3 = 1000^1$
megabajt	MB	$10^6 = 1000^2$
gigabajt	GB	$10^9 = 1000^3$
terabajt	TB	$10^{12} = 1000^4$
petabajt	PB	$10^{15} = 1000^5$
eksabajt	EB	$10^{18} = 1000^6$
zettabajt	ZB	$10^{21} = 1000^7$
jottabajt	YB	$10^{24} = 1000^8$

Przedrostki binarne (IEC 60027-2)		
Nazwa	Symbol	Mnożnik
bajt	B	---
kibibajt	KiB	$2^{10} = 1024^1$
mebibajt	MiB	$2^{20} = 1024^2$
gibibajt	GiB	$2^{30} = 1024^3$
tebibajt	TiB	$2^{40} = 1024^4$
pebibajt	PiB	$2^{50} = 1024^5$
eksbibajt	EiB	$2^{60} = 1024^6$
zebibajt	ZiB	$2^{70} = 1024^7$
jobibajt	YiB	$2^{80} = 1024^8$

Jednostki informacji - bajt

- Przedrostki binarne (dwójkowe) nie zostały przyjęte przez wszystkie środowiska zajmujące się informatyką
- Producenci nośników pamięci korzystają z przedrostków dziesiętnych

Prefiks	Nazwa	System SI	System binarny	Różnica
k	kilo	$10^3 = 1000$	$2^{10} = 1024$	2,40%
M	mega	$10^6 = 1\,000\,000$	$2^{20} = 1\,048\,576$	4,86%
G	giga	$10^9 = 1\,000\,000\,000$	$2^{30} = 1\,073\,741\,824$	7,37%
T	tera	$10^{12} = 1\,000\,000\,000\,000$	$2^{40} = 1\,099\,511\,627\,776$	9,95%

- Z ulotki „Dysk Desktop HDD - zestawienie danych”, Seagate:
 - w przypadku oznaczania pojemności dysków, jeden gigabajt (oznaczany także jako „GB”) jest równy jednemu miliardowi bajtów, a jeden terabajt (oznaczany także jako „TB”) jest równy jednemu bilionowi bajtów

Jednostki informacji - bajt

- Seagate ST1000DM003 (1 TB)
- Drive specification:
 - formatted capacity: 1000 GB (1 TB)
 - guaranteed sectors: 1,953,525,168
 - bytes per sector: 4096
(4K physical emulated at 512-byte sectors)



- Pojemność dysku:
 - $1.953.525.168 \times 512 = 1.000.204.886.016$ bajtów
 - $1.000.204.886.016 / (1024) = 976.762.584$ kB
 - $1.000.204.886.016 / (1024 \times 1024) = 953.870$ MB
 - $1.000.204.886.016 / (1024 \times 1024 \times 1024) = 931,5$ GB

Słowo maszynowe (słowo)

- **Słowo maszynowe** (słowo - ang. word) - jednostka danych używana przez określony komputer (określoną architekturę)
- Słowo składa się odgórnie określonej liczby bitów, nazywanej **długością** lub **szerokością słowa** (najczęściej jest to potęga 2, np. 8, 16, 32, 64 bity)
- Zazwyczaj wielkość słowa określa:
 - rozmiar rejestrów procesora
 - rozmiar szyny danych i szyny adresowej
- Architektury:
 - 8-bitowa: Intel 8080, Z80, Motorola 6800, Intel 8051
 - 16-bitowa: Intel 8086, Intel 80286
 - 32-bitowa: Intel od 80386 do i7, AMD od 5x86 do Athlona, ARM
 - 64-bitowa: Intel Itanium, Pentium 4/EM64T, Core 2, Core i7
AMD Opteron, Athlon 64, Athlon II

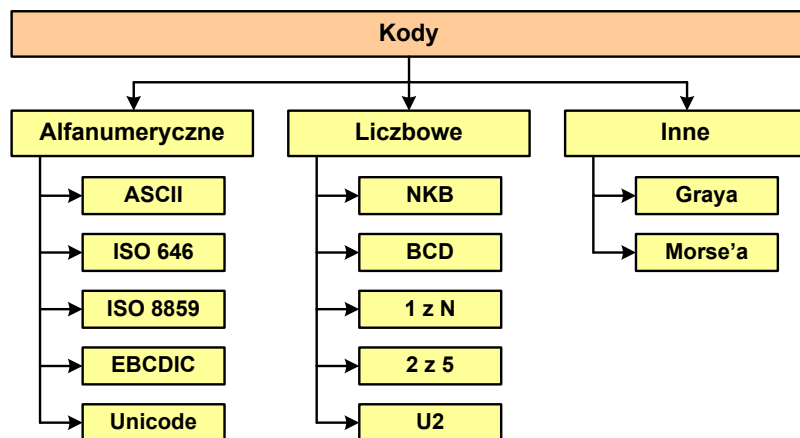
FLOPS

- **FLOPS** (FLoating point Operations Per Second)
 - liczba operacji zmiennoprzecinkowych na sekundę
 - jednostka wydajności układów zmiennoprzecinkowych
- Przykłady wydajności procesorów (teoretyczne):
 - Intel Core i7 975 3,46 GHz - 55,36 GFlops
 - Intel Core2 Quad Q9650 3,00 GHz - 48 GFlops
 - Intel Core2 Duo E8400 3,00 GHz - 24 GFlops
 - najszybszy system równoległy na świecie:
Supercomputer Fugaku (Japan) - 442.010.000 GFlops
RIKEN Center for Computational Science
processors: Fujitsu A64FX 2.2 GHz
nodes: 158.976, cores: 7.630.848
Red Hat Enterprise Linux
www.top500.org



Kodowanie

- **Kodowanie** - proces przekształcania jednego rodzaju postaci informacji na inną postać



Kod ASCII - Kody sterujące

- Kody sterujące - 33 kody, o numerach: 0-31, 127

Dec Hex Char	Dec Hex Char
0 0 NUL (null)	16 10 DLE (data link escape)
1 1 SOH (start of heading)	17 11 DC1 (device control 1)
2 2 STX (start of text)	18 12 DC2 (device control 2)
3 3 ETX (end of text)	19 13 DC3 (device control 3)
4 4 EOT (end of transmission)	20 14 DC4 (device control 4)
5 5 ENQ (enquiry)	21 15 NAK (negative acknowledge)
6 6 ACK (acknowledge)	22 16 SYN (synchronous idle)
7 7 BEL (bell)	23 17 ETB (end of trans. block)
8 8 BS (backspace)	24 18 CAN (cancel)
9 9 TAB (horizontal tab)	25 19 EM (end of medium)
10 A LF (NL line feed, new line)	26 1A SUB (substitute)
11 B VT (vertical tab)	27 1B ESC (escape)
12 C FF (NP form feed, new page)	28 1C FS (file separator)
13 D CR (carriage return)	29 1D GS (group separator)
14 E SO (shift out)	30 1E RS (record separator)
15 F SI (shift in)	31 1F US (unit separator)
	127 7F DEL

- W języku C:

0 (NULL) - \0 7 (BEL) - \a 8 (BS) - \b
9 (TAB) - \t 10 (LF) - \n 13 (CR) - \r

Kod ASCII

- **ASCII - American Standard Code for Information Interchange**

- 7-bitowy kod przypisujący liczby z zakresu 0-127:
 - literom (alfabet angielski)
 - cyfrom
 - znakom przestankowym
 - innym symbolom
 - poleceniom sterującym.

Dec Hex Char	Dec Hex Char	Dec Hex Char	Dec Hex Char
0 0 NUL	32 20 Space	64 40 @	96 60 `
1 1 SOH	33 21 !	65 41 A	97 61 a
2 2 STX	34 22 "	66 42 B	98 62 b
3 3 ETX	35 23 #	67 43 C	99 63 c
4 4 EOT	36 24 \$	68 44 D	100 64 d
5 5 ENQ	37 25 %	69 45 E	101 65 e
6 6 ACK	38 26 &	70 46 F	102 66 f
7 7 BEL	39 27 ^	71 47 G	103 67 g
8 8 BS	40 28 (72 48 H	104 68 h
9 9 TAB	41 29)	73 49 I	105 69 i
10 A LF	42 2A *	74 4A J	106 6A j
11 B VT	43 2B +	75 4B K	107 6B k
12 C FF	44 2C ,	76 4C L	108 6C l
13 D CR	45 2D -	77 4D M	109 6D m
14 E SO	46 2E .	78 4E N	110 6E n
15 F SI	47 2F /	79 4F O	111 6F o
16 10 DLE	48 30 0	80 50 P	112 70 p
17 11 DC1	49 31 1	81 51 Q	113 71 q
18 12 DC2	50 32 2	82 52 R	114 72 r
19 13 DC3	51 33 3	83 53 S	115 73 s
20 14 DC4	52 34 4	84 54 T	116 74 t
21 15 NAK	53 35 5	85 55 U	117 75 u
22 16 SYN	54 36 6	86 56 V	118 76 v
23 17 ETB	55 37 7	87 57 W	119 77 w
24 18 CAN	56 38 8	88 58 X	120 78 x
25 19 EM	57 39 9	89 59 Y	121 79 y
26 1A SUB	58 3A :	90 5A Z	122 7A z
27 1B ESC	59 3B ;	91 5B [123 7B {
28 1C FS	60 3C <	92 5C \	124 7C
29 1D GS	61 3D =	93 5D]	125 7D }
30 1E RS	62 3E >	94 5E ^	126 7E ~
31 1F US	63 3F ?	95 5F _	127 7F DEL

Kod ASCII - Pliki tekstowe

- Elementami pliku tekstowego są **wiersze**, mogą one mieć różną długość
- W systemie Windows każdy wiersz pliku zakończony jest parą znaków:
 - CR, ang. carriage return - powrót karetki, kod ASCII - 13₍₁₀₎ = 0D₍₁₆₎
 - LF, ang. line feed - przesunięcie o wiersz, kod ASCII - 10₍₁₀₎ = 0A₍₁₆₎

- Załóżmy, że plik tekstowy ma postać:

Pierwszy wiersz pliku
Drugi wiersz pliku
Trzeci wiersz pliku

- Rzeczywista zawartość pliku jest następująca:

```

00000000: 50 69 65 72 77 73 7A 79|20 77 69 65 72 73 7A 20 | Pierwszy wiersz
00000010: 70 6C 69 68 75 0D 0A 44|72 75 67 69 20 77 69 65 | plikuDrugi wie
00000020: 72 73 7A 20 70 6C 69 68|75 0D 0A 54 72 7A 65 63 | rsz plikuTrzec
00000030: 69 20 77 69 65 72 73 7A|20 70 6C 69 68 75 0D 0A | i wiersz pliku
    
```

- Wydruk zawiera:

- przesunięcie od początku pliku (szesnastkowo)
- wartości poszczególnych bajtów pliku (szesnastkowo)
- znaki odpowiadające bajtom pliku (traktując bajty jako kody ASCII)

Kod ASCII - Pliki tekstowe

- W systemie Linux znakiem końca wiersza jest tylko LF o kodzie ASCII - $10_{(10)} = 0A_{(16)}$

- Załóżmy, że plik tekstowy ma postać:

Pierwszy wiersz pliku
Drugi wiersz pliku
Trzeci wiersz pliku

- Rzeczywista zawartość pliku jest następująca:

```
00000000: 50 69 65 72 77 73 7a 79|20 77 69 65 72 73 7a 20 | Pierwszy wiersz
00000010: 70 6c 69 68 75 0a 44 72|75 67 69 20 77 69 65 72 | plikuDrugi wiersz
00000020: 73 7a 20 70 6c 69 68 75 0a 54 72 7a 65 63 69 20 | sz plikuTrzeci
00000030: 77 69 65 72 73 7a 20 70|6c 69 68 75 0a | wiersz pliku
```

- Podczas przesyłania pliku tekstowego (np. przez protokół ftp) z systemu Linux do systemu Windows pojedynczy znak LF zamieniany jest automatycznie na parę znaków CR i LF
- Błędne przesłanie pliku tekstowego (w trybie binarnym) powoduje nieprawidłowe jego wyświetlanie:

Pierwszy wiersz plikuDrugi wiersz plikuTrzeci wiersz pliku

ISO/IEC 646

- ISO/IEC 646 - norma definiująca modyfikację 7-bitowego kodowania ASCII, stosowana w latach 70-tych i 80-tych
- W normie określono 10 pozycji na znaki w języku kraju, który przyjął tę normę oraz 2 pozycje na znaki walut

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	SP	!	"	#	\$	%	&	\	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

- zółty - znaki narodowe
- niebieski - znaki walut

- Wszystkie pozostałe znaki są zgodne z ASCII

ISO/IEC 646 - odmiany narodowe

USA

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	SP	!	"	#	\$	%	&	\	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Niemcy

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	SP	!	"	#	\$	%	&	\	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	Š	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	

Francja

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	SP	!	"	#	\$	%	&	\	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	À	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	Ç	Š	^	_	
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	é	ù	ÿ	~	

Polska

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	SP	!	"	#	zł	%	&	\	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	ę	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	ż	ń	ś	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	ó	ł	ź	ć	

ISO/IEC 8859

- ISO/IEC 8859 - zestaw standardów służących do kodowania znaków za pomocą 8-bitów
- Wszystkie zestawy ISO 8859 mają znaki $0_{(10)}-127_{(10)}$ ($00_{(16)}-7F_{(16)}$) takie same jak w kodzie ASCII
- Pozycjom $128_{(10)}-159_{(10)}$ ($80_{(16)}-9F_{(16)}$) przypisane są dodatkowe kody sterujące, tzw. C1 (obecnie nie są używane)
- Od czerwca 2004 roku ISO 8859 nie jest rozwijane.

ISO/IEC 8859

■ Stosowane standardy ISO 8859:

- ISO 8859-1 (Latin-1) - alfabet łaciński dla Europy zachodniej
- ISO 8859-2 (Latin-2) - łaciński dla Europy środkowej i wschodniej
- ISO 8859-3 (Latin-3) - łaciński dla Europy południowej
- ISO 8859-4 (Latin-4) - łaciński dla Europy północnej
- ISO 8859-5 (Cyrillic) - dla cyrylicy
- ISO 8859-6 (Arabic) - dla alfabetu arabskiego
- ISO 8859-7 (Greek) - dla alfabetu greckiego
- ISO 8859-8 (Hebrew) - dla alfabetu hebrajskiego
- ISO 8859-9 (Latin-5)
- ISO 8859-10 (Latin-6)
- ISO 8859-11 (Thai) - dla alfabetu tajskiego
- ISO 8859-12 - brak
- ISO 8859-13 (Latin-7)
- ISO 8859-14 (Latin-8) - zawiera polskie litery
- ISO 8859-15 (Latin-9)
- ISO 8859-16 (Latin-10) - łaciński dla Europy środkowej, zawiera polskie litery

ISO/IEC 8859-2

- ISO/IEC 8859-2, Latin-2 („środkowo”, „wschodnioeuropejskie”)
- dostępne języki: bośniacki, chorwacki, czeski, węgierski, polski, rumuński, serbski, serbsko-chorwacki, słowacki, słoweński, górno- i dolnołużycki
- możliwość przedstawienia znaków w języku niemieckim i angielskim
- 191 znaków łacińskiego pisma
- do 02.11.2015 kodowanie to było zgodne z **Polską Normą**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10																
20	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Nie używane															
90																
A0	NB	À	Á	Â	Ã	Ä	Å	Ā	Ă	Ą	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ
B0	°	à	á	â	ã	ä	å	ā	ă	ą	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ
C0	Ř	Ǻ	ǻ	Ǽ	Ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ
D0	Đ	Ñ	Ń	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Û	Ü	Ý	Ț	Ț
E0	đ	ñ	ń	ó	ô	õ	ö	×	ř	ů	ú	û	ü	ý	ț	ț
F0	đ	ñ	ń	ó	ô	õ	ö	÷	ř	ů	ú	û	ü	ý	ț	ț

SP - spacja
NBSP - twarda spacja
SHY - miękki dywiz (myślnik)

ISO/IEC 8859-1

- ISO/IEC 8859-1, Latin-1 („zachodnioeuropejskie”)
- kodowanie używane w Amerykach, Europie Zachodniej, Oceanii i większej części Afryki
- dostępne języki: albański, angielski, baskijski, duński, estoński, fiński, francuski, hiszpański, irlandzki, islandzki, kataloński, łaciński, niderlandzki, niemiecki, norweski, portugalski, retoromański, szkocki, szwedzki, włoski
- 191 znaków łacińskiego pisma.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10																
20	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Nie używane															
90																
A0	NB	À	Á	Â	Ã	Ä	Å	Ā	Ă	Ą	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ
B0	°	à	á	â	ã	ä	å	ā	ă	ą	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ
C0	Ř	Ǻ	ǻ	Ǽ	Ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ
D0	Đ	Ñ	Ń	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Û	Ü	Ý	Ț	Ț
E0	đ	ñ	ń	ó	ô	õ	ö	×	ř	ů	ú	û	ü	ý	ț	ț
F0	đ	ñ	ń	ó	ô	õ	ö	÷	ř	ů	ú	û	ü	ý	ț	ț

SP - spacja
NBSP - twarda spacja
SHY - miękki dywiz (myślnik)

ISO/IEC 8859-2 - Litery diakrytyczne w j. polskim

- 18 liter:
- Ą - ą
- Ć - ć
- Ę - ę
- Ł - ł
- Ń - ń
- Ó - ó
- Ś - ś
- Ź - ź
- Ż - ż

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10																
20	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Nie używane															
90																
A0	NB	Ą	Ć	Ę	Ł	Ń	Ś	Ś	Ś	Ț	Ț	Ț	Ț	Ț	Ț	Ț
B0	°	ą	ć	ę	ł	ń	ś	ś	ś	ț	ț	ț	ț	ț	ț	ț
C0	Ř	Ǻ	ǻ	Ǽ	Ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ	ǿ
D0	Đ	Ñ	Ń	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Û	Ü	Ý	Ț	Ț
E0	đ	ñ	ń	ó	ô	õ	ö	×	ř	ů	ú	û	ü	ý	ț	ț
F0	đ	ñ	ń	ó	ô	õ	ö	÷	ř	ů	ú	û	ü	ý	ț	ț

ISO/IEC 8859-1 i ISO/IEC 8859-2 - porównanie

ISO 8859-1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0	NB SP	í	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	—
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

ISO 8859-2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0	NB SP	À	˘	Ĺ	š	Š	Ť	Ž	SHY	Ž	Ž					
B0	°	á	‚	ĺ	ı	š	š	ť	ž	~	ž	ž				
C0	Ř	Ā	Ă	Ä	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	ř	á	â	ă	ä	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā
F0	đ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

EBCDIC

- **EBCDIC - Extended Binary Coded Decimal Interchange Code**
- 8-bitowe kodowanie znaków stworzone jako rozszerzenie kodowania BCD

- używane głównie w systemach IBM w latach 60-tych XX wieku
- umożliwia zapisanie do 256 różnych symboli
- brak zachowania kolejności liter zgodnie z kolejnością kodów, np. po R nie ma S
- kody EBCDIC **nie są zgodne** z ASCII.

EBCDIC

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	Znaki kontrolne															
30	Znaki kontrolne															
40	SP	NB	SP	à	â	ã	ä	å	æ	ç	ñ	[.	<	(+
50	&	é	ê	ë	è	í	î	ï	ì	í	î	ï	ì	í	î	ï
60	-	/	À	Á	Â	Ã	Ä	Å	Æ	Ç	Ñ		,	%	_	>
70	ø	È	É	Ê	Ë	Ì	Í	Î	Ï	Ì	Í	Î	Ï	Ì	Í	Î
80	ø	a	b	c	d	e	f	g	h	i	<<	>>	ø	Ý	Þ	±
90	°	j	k	l	m	n	o	p	q	r	ª	º	æ	.	Æ	¤
A0	µ	~	s	t	u	v	w	x	y	z	ı	đ	Ý	þ	®	
B0	ç	£	¥	·	©	§	¶	¼	½	¾	¬		—	‘	/	x
C0	{	A	B	C	D	E	F	G	H	I	SHY	ó	ô	õ	ö	ø
D0	}	J	K	L	M	N	O	P	Q	R	¹	ú	û	ü	ý	ÿ
E0	\	÷	S	T	U	V	W	X	Y	Z	²	ó	ô	õ	ö	ø
F0	0	1	2	3	4	5	6	7	8	9	ª	º	¼	½	¾	ø

EBCDIC i ISO 8859-1 - porównanie

ISO 8859-1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	SP	!	"	#	\$	%	&	\	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Nieużywane															
90	Nieużywane															
A0	NB	SP	í	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

EBCDIC

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
10	Znaki kontrolne															
20	Znaki kontrolne															
30	Znaki kontrolne															
40	SP	NB	SP	à	â	ã	ä	å	æ	ç	ñ	[.	<	(+
50	&	é	ê	ë	è	í	î	ï	ì	í	î	ï	ì	í	î	ï
60	-	/	À	Á	Â	Ã	Ä	Å	Æ	Ç	Ñ		,	%	_	>
70	ø	È	É	Ê	Ë	Ì	Í	Î	Ï	Ì	Í	Î	Ï	Ì	Í	Î
80	ø	a	b	c	d	e	f	g	h	i	<<	>>	ø	Ý	Þ	±
90	°	j	k	l	m	n	o	p	q	r	ª	º	æ	.	Æ	¤
A0	µ	~	s	t	u	v	w	x	y	z	ı	đ	Ý	þ	®	
B0	ç	£	¥	·	©	§	¶	¼	½	¾	¬		—	‘	/	x
C0	{	A	B	C	D	E	F	G	H	I	SHY	ó	ô	õ	ö	ø
D0	}	J	K	L	M	N	O	P	Q	R	¹	ú	û	ü	ý	ÿ
E0	\	÷	S	T	U	V	W	X	Y	Z	²	ó	ô	õ	ö	ø
F0	0	1	2	3	4	5	6	7	8	9	ª	º	¼	½	¾	ø

Windows-1250

- **Windows-1250 (CP-1250)** - strona kodowa używana przez system Microsoft Windows do reprezentacji tekstów w językach środkowoeuropejskich używających alfabetu łacińskiego
- Obsługiwane języki: albański, chorwacki, czeski, polski, rumuński, słowacki, słoweński, węgierski (ale także niemiecki)
- Windows-1250 jest podobny do ISO 8859-2 - posiada wszystkie jego drukowalne znaki (a także kilka dodatkowych), lecz kilka z nich zajmuje inne miejsca.

ISO 8859-2 i Windows-1250 - porównanie

ISO 8859-2																Windows-1250																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne																Znaki kontrolne																
10																																	
20	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~		p	q	r	s	t	u	v	w	x	y	z	{		}	~		
80	Nieużywane																																
90																																	
A0	NBS	À	Á	Â	Ã	Ä	Å	Ā	Ă	Ą	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ	NBS	À	Á	Â	Ã	Ä	Å	Ā	Ă	Ą	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ	Ȧ	
B0	°	à	á	â	ã	ä	å	ā	ă	ą	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ	°	à	á	â	ã	ä	å	ā	ă	ą	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ	
C0	Ř	Š	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ř	Š	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	Ț	
D0	Đ	Ñ	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Đ	Ñ	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	
E0	í	á	â	ã	ä	å	ā	ă	ą	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ	í	á	â	ã	ä	å	ā	ă	ą	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ	ȧ	
F0	đ	ñ	ń	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	đ	ñ	ń	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	ó	

Problem kodowania polskich liter diakrytycznych

■ Problem z wyświetlaniem polskich liter diakrytycznych

- Tekst zapisany w standardzie ISO-8859-2:

Ą Ć Ę Ł Ń Ó Ś Ż Ż
ą ć ę ł ń ó ś ż ż

- Tekst wyświetlony w Notatniku systemu Windows (Windows-1250):

Ć Ę Ł Ń Ó ĩ Ń Ż
ą ć ę ł ń ó ł Ń ż

Unicode (Unikod)



- Komputerowy zestaw znaków mający obejmować wszystkie pisma i inne znaki (symbole techniczne, wymowy) używane na świecie
- Unicode przypisuje unikalny numer każdemu znakowi, niezależny od używanej platformy, programu czy języka
- Rozwijany przez konsorcjum utworzone przez firmy komputerowe, producentów oprogramowania oraz grupy użytkowników
 - <http://www.unicode.org>
- Pierwsza wersja: **Unicode 1.0** (10.1991)
- Ostatnia wersja: **Unicode 14.0** (14.09.2021)
 - The Unicode Consortium. The Unicode Standard, Version 14.0.0, (Mountain View, CA: The Unicode Consortium, 2021)
 - <http://www.unicode.org/versions/Unicode14.0.0/>
 - koduje 144.697 znaków

Unicode - Zakresy



Zakres:	Znaczenie:
U+0000 - U+007F	Basic Latin (to samo co w ASCII)
U+0080 - U+00FF	Latin-1 Supplement (to samo co w ISO/IEC 8859-1)
U+0100 - U+017F	Latin Extended-A
U+0180 - U+024F	Latin Extended-B
U+0250 - U+02AF	IPA Extensions
U+02B0 - U+02FF	Spacing Modifiers Letters
...	
U+0370 - U+03FF	Greek
U+0400 - U+04FF	Cyrillic
...	
U+1D00 - U+1D7F	Phonetic Extensions
U+1D80 - U+1DBF	Phonetic Extensions Supplement
U+1E00 - U+1EFF	Latin Extended Additional
U+1F00 - U+1FFF	Greek Extended
...	

Unicode



- Standard Unicode definiuje nie tylko kody numeryczne przypisane poszczególnym znakom, ale także określa sposób bajtowego **kodowania** znaków
- Kodowanie określa sposób w jaki znaki ze zbioru mają być zapisane w **postaci binarnej**
- Istnieją trzy podstawowe metody kodowania:
 - 32-bitowe: UTF-32
 - 16-bitowe: UTF-16
 - 8-bitowe: UTF-8gdzie: **UTF** - UCS Transformation Format
UCS - Universal Character Set
- Wszystkie metody obejmują wszystkie kodowane znaki w Unicode.

Unicode



- Metody kodowania różnią się liczbą bajtów przeznaczonych do opisanego kodu znaku

A 00000041	Ω 000003A9	語 00008A9E	𐄆 00010384	UTF-32
A 0041	Ω 03A9	語 8A9E	𐄆 D800 DF84	UTF-16
A 41	Ω CE A9	語 E8 AA 9E	𐄆 F0 90 8E 84	UTF-8

źródło: The Unicode Consortium. The Unicode Standard, Version 8.0

Unicode - kodowanie UTF-32



- UTF-32** - sposób kodowania standardu Unicode wymagający użycia 32-bitowych słów

A 00000041	Ω 000003A9	語 00008A9E	𐄆 00010384	UTF-32
---------------	---------------	---------------	---------------	--------

- Kod znaku ma zawsze stałą długość 4 bajtów i przedstawia numer znaku w tabeli Unikodu
- Kody obejmują zakres od 0 do 0x10FFFF (od 0 do 1 114 111)
- Kodowanie to jest jednak bardzo nieefektywne - zakodowane ciągi znaków są 2-4 razy dłuższe niż ciągi tych samych znaków zapisanych w innych kodowaniach.

Unicode - kodowanie UTF-16



- UTF-16** - sposób kodowania standardu Unicode wymagający użycia 16-bitowych słów

A 0041	Ω 03A9	語 8A9E	𐄆 D800 DF84	UTF-16
-----------	-----------	-----------	----------------	--------

- Dla znaków z przedziału od U+0000 do U+FFFF używane jest jedno słowo, którego wartość jest jednocześnie kodem znaku w Unicode
- Dla znaków z wyższych pozycji używa się dwóch słów:
 - pierwsze słowo należy do przedziału: U+D800 - U+DBFF
 - drugie słowo należy do przedziału: U+DC00 - U+DFFF.

Unicode - kodowanie UTF-8



- UTF-8 - kodowanie ze zmienną długością reprezentacji znaku wymagające użycia 8-bitowych słów

A	Ω	語	Ⅲ
41	CE A9	E8 AA 9E	F0 90 8E 84

UTF-8

- Znaki Unikodu są mapowane na ciągi bajtów
 - 0x00 do 0x7F - bity 0xxxxxxx
 - 0x80 do 0x7FF - bity 110xxxxx 10xxxxxx
 - 0x800 do 0xFFFF - bity 1110xxxx 10xxxxxx 10xxxxxx
 - 0x10000 do 0x1FFFFF - bity 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
 - 0x200000 do 0x3FFFFFFF - bity 111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
 - 0x4000000 do 0x7FFFFFFF - bity 1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Unicode



27308

CJK Unified Ideographs Extension B

27342

27308 虫 142.8 𧈧 𧈨 𧈩 UCS2003 GKX-1086.03 T4-4721	2731B 虫 142.8 𧈪 𧈫 𧈬 UCS2003 GKX-1088.15 T6-617B	2732F 虫 142.8 𧈯 𧈰 UCS2003 GHC
27309 虫 142.8 𧈭 𧈮 𧈯 UCS2003 GKX-1086.05 T5-4955	2731C 虫 142.8 𧈱 𧈲 𧈳 UCS2003 GKX-1088.16 T6-6221	27330 虫 142.9 𧈴 𧈵 UCS2003 GHC
2730A 虫 142.8 𧈷 𧈸 𧈹 UCS2003 GKX-1086.08 T4-467D	2731D 虫 142.8 𧈻 𧈼 𧈽 UCS2003 GKX-1088.17 T5-4960	27331 虫 142.8 𧈾 𧈿 UCS2003 G4K
2730B 虫 142.8 𧈿 𧉀 𧉁 UCS2003 GKX-1086.10 T6-6223	2731E 虫 142.7 𧉂 𧉃 UCS2003 GKX-1088.18	27332 虫 142.8 𧉄 𧉅 UCS2003 GHC
2730C 虫 142.8 𧉆 𧉇 𧉈 UCS2003 GKX-1086.12 T5-495F	2731F 虫 142.8 𧉉 𧉊 𧉋 UCS2003 GKX-1088.19 T6-6174	27333 虫 142.8 𧉌 𧉍 UCS2003 GHC
2730D 虫 142.8 𧉎 𧉏 𧉐 UCS2003 GKX-1086.22 T4-4677	27320 虫 142.8 𧉑 𧉒 𧉓 UCS2003 GKX-1088.20 T6-6170	27334 虫 142.8 𧉔 𧉕 UCS2003 T5-4953

Unicode



	010	011	012	013	014	015	016	017
0	Ā	Đ	Ġ	Ī	Ĳ	Ō	Š	Ū
1	ā	đ	ġ	ī	ĳ	ō	š	ū
2	Ă	Ĕ	Ģ	Ĳ	Ĳ	Ţ	Ũ	
3	ă	ĕ	ģ	ĳ	ĳ	ţ	ũ	
4	Ą	Ě	Ĥ	Ĵ	ń	Ŕ	Ť	Ŵ
5	ą	ě	ĥ	ĵ	ņ	ŕ	ť	ŵ
6	Ć	Ę	Ħ	Ķ	ņ	Ŗ	Ŧ	Ŷ
7	ć	ę	ħ	ķ	ņ	ŗ	ŧ	ŷ

European Latin

- 0100 Ā LATIN CAPITAL LETTER A WITH MACRON ≡ 0041 A 0304 5
- 0101 ā LATIN SMALL LETTER A WITH MACRON • Latvian, Latin, ... ≡ 0061 a 0304 5
- 0102 Ă LATIN CAPITAL LETTER A WITH BREVE ≡ 0041 A 0306 8
- 0103 ă LATIN SMALL LETTER A WITH BREVE • Romanian, Vietnamese, Latin, ... ≡ 0061 a 0306 8
- 0104 Ą LATIN CAPITAL LETTER A WITH OGONEK ≡ 0041 A 0328 9
- 0105 ą LATIN SMALL LETTER A WITH OGONEK • Polish, Lithuanian, ... ≡ 0061 a 0328 9
- 0106 Ć LATIN CAPITAL LETTER C WITH ACUTE ≡ 0043 C 0301 6
- 0107 ć LATIN SMALL LETTER C WITH ACUTE • Polish, Croatian, ... → 045B ħ cyrillic small letter tshe ≡ 0063 c 0301 6

Koniec wykładu nr 4

Dziękuję za uwagę!