



## Základy počítačů

Prednášky (povinné):

- Ing. Richard Balogh, PhD.
- Ing. Štefan Chamraz, PhD.
- doc. Ing. Katarína Žáková, PhD. - garant

Zimný semester

Počet kreditů 6.0 (6 ECTS = 150 hod. práce)

Prednáška / Cvičenie: **2.0 / 2.0** [hod./týždeň]

+ **samoštúdium**: 4.0 – 6.0 [hod./týždeň] !!

## Čo sú data?

46 45 49<sup>HEX</sup>

0100 0110 0100 0101 0100 1001<sup>BIN</sup>

**FEI**  
ASCII

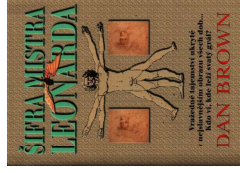
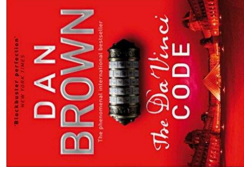
00000000 46 INC ESI ; inkrementuj ESI register  
00000001 45 INC EBP ; inkrementuj EBP register  
00000002 49 DEC ECX ; dekrementuj ECX register

**4 605 257 €**

html color



## Kódy a šifry



## Kódy a šifry

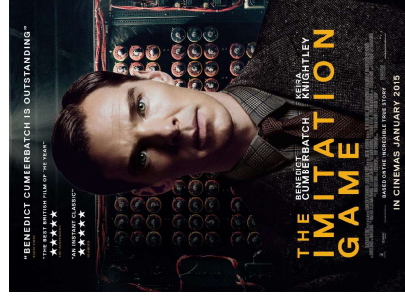
[en] Code  
[de] Kode  
[ru] Код  
[sk,cz] Kód

- + súbor znakov určených na zaznamenávanie informácií určitým spôsobom a pravidlá na ich používanie
- + predpis na prevádzkanie jednej sústavy znakov do inej

[en] Cipher  
[de] Verschlüsselungsverfahren  
[sk] Šifra  
[ru] Шифр

- + kryptografický algoritmus, ktorý transformuje čitateľnú správu (prostý text) do nečitateľnej podoby (šifrovy text).

## Imitation game / Enigma







# Unicode

http://www.unicode.org/

Štandard kódovania znakov vyvinutý organizáciou Unicode Consortium (1991)

Paralelne ISO 10646 – definuje UCS (Universal Character Set) – došlo k zjednoteniu kódovania znakov

snaha obsiahnuť takmer všetky znaky používané v jazykoch na svete (japončina, čínština – tisíce znakov)

prvých 256 znakov je zhodných s rozšíreným ASCII pôvodne bol pre každý znak pridelený 16-bitový jedinečný kód v súčasnosti 31-bitové, štandard je navrhnutý tak, že všetky možné znaky rozdeľuje do sedemnástich skupín po 65536 znakov. To znamená môžeme definovať až 1 114 112 (  $17 \times 2^{16}$  ) znakov.

**UTF-8 :** Kóduje do 1 až 6 bajtov; Do 1 bajtu kóduje prvých 128 znakov zo znakovej sady US-ASCII ( U+0000 až U+007F);

| Unicode hodnota          | UTF-8 sekvencia   |
|--------------------------|---|
| U-00000000 - U-0000007F  | 0xxxxxxx<br>Znaky s číslami od 0 do 127 sa kódujú ako jeden bajt  |
| U-00000080 - U-000007FF  | 110xxxxx 10xxxxxx<br>Znaky s číslami od 128 do 2 047 sa kódujú ako dva bajty                            |
| U-00000800 - U-0000FFFF  | 1110xxxx 10xxxxxx 10xxxxxx<br>Znaky s číslami od 2 048 do 65 535 sa kódujú ako tri bajty                |
| U-00010000 - U-001FFFFF  | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx<br>Znaky s číslami od 65 536 do 1 114 111 sa kódujú ako štyri bajty |
| U-00200000 - U-03FFFFFF  | 111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx<br>U-040000000 - U-7FFFFFFF                                |
| U-040000000 - U-7FFFFFFF | 1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx   |

Staršie webové prehliadače (**HTML** dokumenty) podporovali len znakovú sadu ASCII, moderné majú prednastavenú sadu ISO-8859-1, niek musia mať v položke <meta> uvedenú znakovú sadu buď ISO-8858-xx, UTF8 alebo UTF16. Podobné princípy sú uplatňované v XML (eXtensible Markup Language) dokumentoch, ktoré okrem iného slúžia na výmenu údajov medzi aplikáciami a zverejňovanie dokumentov.

## Dátum

**Europe :** little - endian  
Day, Month, Year (24<sup>th</sup>, May, 2006 = 24/ 5/2006)

**US :** middle - endian  
Month, Day, Year (May, 24<sup>th</sup>, 2006 = 5/24/2006)

**China, Japan & ISO 8601: big - endian**  
Year, Month, Day (2006, May, 24<sup>th</sup> = 2006-05-24)

### Poznámky k endianom:

Nedá sa povedať, že niektorý endian je výhodnejší voči inému, len ak zapíšeme dátum v big endian – ľahšie sa trieda položky. Ak prenášame súbory medzi počítačmi s rôznymi endiami, treba robiť vykonat transformáciu.

Štandard Unicode sa oproti ISO 10646 navyše zaoberá

algoritmami pre písmo písané smerom doľava (arabština) podporou obojstranných textov (ako napr. zmes hebrejštiny a latiniky) algoritmy pre usporiadanie a porovnávanie textov

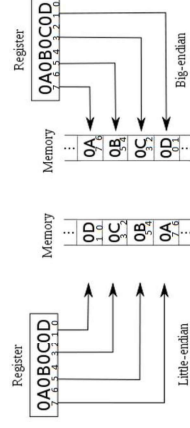
Nevýhodou je dĺžka znakov, prítomnosť kódov v znakových ako nulový bajt, znak “\0”, atď.

Vznik systémov kódovania UTF (Unicode Transformation Format ) ako sú UTF-8, UTF-16 a UTF-32.

Väčšina rozhraní systému Windows používa formu UTF-16.

## Ukladanie viacbajtových údajov do pamäte

Problém ukladania informácií je, ak veľkosť údajového typu nie je zhodná s veľkosťou strojového slova.



Niekedy sa tento problém nazýva aj NUXI problém . (refazer, „UNIX“ môže počítať s opätou orientáciou zobrazit' ako „NUXI“)



## Aritmetika: násobenie

Vypočítame súčin 3 x 6 v dvojkovej sústave:

$$\begin{array}{r} 011 \\ \cdot 110 \\ \hline 100 \\ 1011 \\ \hline 10110 \end{array} = 18_{10}$$

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \\ 0000 \\ \hline 110111 \end{array}$$

## Koľko bitov potrebujeme na zobrazenie daného čísla?

## Racionálne a reálne čísla

Zobrazenie v pevnej rádovej čiarkke

$$g = 9,80665 \text{ m.s}^{-2}$$

počet miest  $m = \log_2 K$

K rôznych objektov □ K rôznych kódov vyjadriteľné pomocou usporiadanej m-tice binárných symbolov

Zobrazenie v pohyblivej rádovej čiarkke

$$M_e = 9,10938356 \times 10^{-31} \text{ kg}$$



### Zobrazenie v pevnej rádovej čiarkke

$$19,75_{10} = 10011,11_2$$

$$19,75 = 1 \cdot 10^1 + 5 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

$1/2 \quad 1/4 \quad 1/8 \quad 1/16 \dots$

1011,11 =

$1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5}$

$1 \cdot 2^1 \quad 0 \cdot 2^0 \quad 0 \cdot 2^{-1} \quad 1 \cdot 2^{-2} \quad 1 \cdot 2^{-3} \quad 1 \cdot 2^{-4} \quad 1 \cdot 2^{-5}$

$2 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$

$1 \quad 1/2 \quad 1/4 \quad 1/8 \quad 1/16 \quad 1/32$

$0,5 \quad 0,25$

### Zobrazenie v pevnej rádovej čiarkke

Příklad: Preveďte číslo  $0,6875_{10}$  do dvojkovej sústavy:

$$\begin{aligned} 0,6875 \times 2 &= 1,3750 \\ 0,3750 \times 2 &= 0,7500 \\ 0,7500 \times 2 &= 1,5000 \\ 0,5000 \times 2 &= 1,0000 \\ 0,0000 \times 2 &= 0,0000 \end{aligned}$$

$0,6875_{10} = 0,1011_2$

$$0,6875_{10} = 0,1011_2$$

Příklad:

Preveďme číslo  $0,1_{10}$  do dvojkovej sústavy

$$\begin{aligned} 0,1000 \cdot 2 &= 0,2000 \\ 0,2000 \cdot 2 &= 0,4000 \\ 0,4000 \cdot 2 &= 0,8000 \\ 0,8000 \cdot 2 &= 1,6000 \\ 0,6000 \cdot 2 &= 1,2000 \\ 0,2000 \cdot 2 &= 0,4000 \\ 0,4000 \cdot 2 &= 0,8000 \\ 0,8000 \cdot 2 &= 1,6000 \\ 0,6000 \cdot 2 &= 1,2000 \end{aligned}$$

Dôsledky:

```
float x;
for ( x = 0; x<=1; x=x+0.1 )
{ /* rob niečo */ }
```

**nekonečný cyklus!**

Správne by malo byť:

```
float x;
for ( x = 0; x<=1; x=x+0.1 )
{ /* rob niečo */ }
```

$$0,1_{10} = 0,000110011_2 = 0,00011_2$$

Číslo  $0,1_{10}$  sa nedá vyjadriť konečným počtom binárných číslic !!

### A čo iné čísla?

- Veľmi veľké čísla (sekundy v storočij)  $3,155,760,000_{10} (3,15576_{10} \cdot 10^9)$
- Veľmi malé čísla (priemer atómu [m])  $0,0000000110 (1,010 \cdot 10^{-8})$
- Racionálne (periódické)  $2/3 (0,66666666666...)$
- Iracionálne čísla  $2^{1/2} (1,414213562373...)$   
 $e (2,718...), \pi (3,141...)$

# Zobrazenie v pohyblivej rádovej čiarkke

Reprezentácia čísel, ktoré sú na vyjadrenie v pevnej rádovej čiarkke

- príliš malé, alebo
- príliš veľké.

Zápis v tvare:

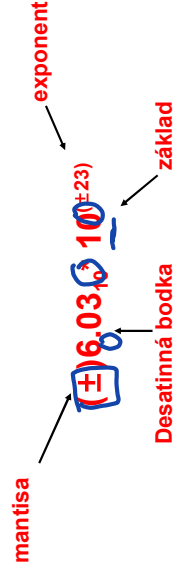
$$M_e = \text{mantisa} \times \text{základ}^{\text{exponent}} = 9,10938356 \times 10^{-31} \text{ kg}$$

Postupne niekoľko rôznych systémov  
Dnes takmer výhradne podľa normy IEEE 754.

# Aritmetika pohyblivej rádovej čiarkky (FPU – Float Point Unit) (FPA -Floating Point Arithmetics)

Presnosť ↔ Rozsah

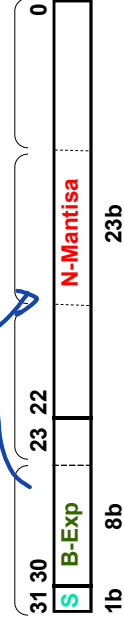
## Vedecké zobrazenie čísel (dekadické)



- Normalizovaný tvar zápisu: bez vodiacich núl (naľavo od desatinnej bodky je len jedna nenulová platná číslica)
- Iný spôsob zápisu:  $1/1\,000\,000\,000$
- Normalizovaný:  $1.0 \cdot 10^{-9}$
- Nenormalizovaný:  $0.1 \cdot 10^{-8}$   
 $10.0 \cdot 10^{-10}$

## Jednoduchá presnosť čísel FP (Single Precision – SP, C: float)

- Formát zápisu:  $(+/-)1.\text{xxxxxxxxxx}_2 \cdot 2^{\text{yyyy}_2}$
- Počet bitov: 32



1b 8b Mantisa: (priamy kód)

S - Sign znamienko mantisy

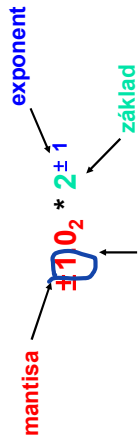
|Mantisa| = 1.xxxxxxxxx, xxxxxxxx = N-Mantisa

Exponent = B-Exp - Bias, Bias = 127, B-Exp = <1, 254>

Číslo z rozsahu:  $2^{-426}(1.0) \sim 2^{+127}(2 - 2^{-23})$

t.j.  $1.18 \cdot 10^{-38} \sim 3.40 \cdot 10^{38}$

## Vedecké zobrazenie čísel (binárne)



„binárna bodka“  
Potrebujeme zapísať:

- „znamienko“ mantisy
  - „znamienko“ exponentu
- Počítateľ podporujú prácu s číslami typu float:  
Forma zápisu znamienka ???

- priamy kód
- jednotkový doplnok
- dvojkový doplnok

## Zobrazenie FP čísel

- Čo sa stane ak je výsledok veľmi veľký? ( $> 3.403 \cdot 10^{38}$ ) **Overflow!**
- Overflow  $\Rightarrow$  Exponent väčší ako sa dá zobrazit' do 8 bitov
- Čo sa stane ak je výsledok veľmi malý? ( $> 0, < 1.17 \cdot 10^{-38}$ ) **Underflow!**
- Underflow  $\Rightarrow$  Záporný exponent "väčší" ako sa dá zobrazit' do 8-bitov

