Reprezentace dat

BI-PA1 – Programování a Algoritmizace I.

Ladislav Vagner

Katedra teoretické informatiky
Fakulta informačních technologií
ČVUT v Praze
xvagner@fit.cvut.cz

2. října 2013

Obsah

- Dvojková (binární) soustava.
- Kódování celých čísel (bez i se znaménkem).
- Kódování desetinných čísel.
- Kódování znaků a řetězců.

Kódování do bajtů a bitů

1. Kolik různých hodnot můžeme uložit do 10 bitů?

Kódování do bajtů a bitů

1. Kolik různých hodnot můžeme uložit do 10 bitů?

$$1024 = 2^{10}$$

2. Kolik bitů je potřeba k uložení celých čísel z uzavřeného intervalu -987654321 až 123456789?

Kódování do bajtů a bitů

1. Kolik různých hodnot můžeme uložit do 10 bitů?

$$1024 = 2^{10}$$

2. Kolik bitů je potřeba k uložení celých čísel z uzavřeného intervalu -987654321 až 123456789?

$$\lceil \log_2(123456789 - (-987654321) + 1) \rceil$$

3. Kolik bitů je potřeba k uložení desetinných čísel z uzavřeného intervalu -123456 až -56 s přesností 0.001?

Kódování do bajtů a bitů

1. Kolik různých hodnot můžeme uložit do 10 bitů?

$$1024 = 2^{10}$$

2. Kolik bitů je potřeba k uložení celých čísel z uzavřeného intervalu -987654321 až 123456789?

$$\lceil \log_2(123456789 - (-987654321) + 1) \rceil$$

3. Kolik bitů je potřeba k uložení desetinných čísel z uzavřeného intervalu -123456 až -56 s přesností 0.001?

$$\lceil \log_2(\frac{-56 - (-123456)}{0.001} + 1) \rceil$$

Dvojková soustava

Převeďte 7964_{dec} do dvojkové soustavy:

`7964 (dec) = ? (bin)`

Dvojková soustava

Převeďte 7964_{dec} do dvojkové soustavy:

```
7964 (dec) = ? (bin)
7964 : 2 = 3982   +0
```

Dvojková soustava

Převeďte 7964_{dec} do dvojkové soustavy:

```
7964 (dec) = ? (bin)
7964 : 2 = 3982   +0
3982 : 2 = 1991   +0
```

Dvojková soustava

Převeďte 7964_{dec} do dvojkové soustavy:

```
7964 (dec) = ? (bin)
7964 : 2 = 3982   +0
3982 : 2 = 1991   +0
1991 : 2 =  995   +1
```

Dvojková soustava

Převeďte 7964_{dec} do dvojkové soustavy:

```
7964 (dec) = ? (bin)
7964 : 2 = 3982   +0
3982 : 2 = 1991   +0
1991 : 2 =  995   +1
 995 : 2 =  497   +1
 497 : 2 =  248   +1
 248 : 2 =  124   +0
 124 : 2 =   62   +0
  62 : 2 =   31   +0
  31 : 2 =   15   +1
  15 : 2 =    7   +1
   7 : 2 =    3   +1
   3 : 2 =    1   +1
   1 : 2 =    0   +1
```

$$7964_{dec} = 1111100011100_{bin} = 1\mathrm{F1C}_{hex}$$

Dvojková soustava

Převeďte $110111010_{bin} = 1BA_{hex}$ do desítkové soustavy:

```
110111010 (bin) = 1BA (hex)
 0 *   1 =    0
 1 *   2 =    2
 0 *   4 =    0
 1 *   8 =    8
 1 *  16 =   16
 1 *  32 =   32
 0 *  64 =    0
 1 * 128 =  128
 1 * 256 =  256
sum         442
```

$$110111010_{bin} = 1BA_{hex} = 442_{dec}$$

Dvojková soustava

Převeďte $11110110110010001010111101011_{bin}$ do šestnáctkové soustavy:

Dvojková soustava

Převeďte $11110110110010001010111101011_{bin}$ do šestnáctkové soustavy:

```
0000 = 0    0001 = 1    0010 = 2    0011 = 3
0100 = 4    0101 = 5    0110 = 6    0111 = 7
1000 = 8    1001 = 9    1010 = A    1011 = B
1100 = C    1101 = D    1110 = E    1111 = F

11110110110010001010111101011 (bin)

0001 1110 1101 1001 0001 0101 1110 1011
   1    E    D    9    1    5    E    B

1ED915EB (hex)
```

Dvojková soustava

Převeďte 168.5685_{dec} do dvojkové soustavy:

Dvojková soustava

Převeďte 168.5685_{dec} do dvojkové soustavy:

```
168 (dec) = ? (bin)
168 : 2 = 84   +0
 84 : 2 = 42   +0
 42 : 2 = 21   +0
 21 : 2 = 10   +1
 10 : 2 =  5   +0
  5 : 2 =  2   +1
  2 : 2 =  1   +0
  1 : 2 =  0   +1
```

$$168_{dec} = 10101000_{bin}$$

Dvojková soustava

Převeďte 168.5685_{dec} do dvojkové soustavy:

```
0.5685 (dec) = ? (bin)
```

Dvojková soustava

Převeďte 168.5685_{dec} do dvojkové soustavy:

```
0.5685 (dec) = ? (bin)
0.5685    * 2 = 1.137
```

Dvojková soustava

Převeďte 168.5685_{dec} do dvojkové soustavy:

```
0.5685 (dec) = ? (bin)
0.5685     * 2 = 1.137
0.137      * 2 = 0.274
```

Dvojková soustava

Převeďte 168.5685_{dec} do dvojkové soustavy:

```
0.5685 (dec) = ? (bin)
0.5685    * 2 = 1.137
0.137     * 2 = 0.274
0.274     * 2 = 0.548
0.548     * 2 = 1.096
0.096     * 2 = 0.192
0.192     * 2 = 0.384
0.384     * 2 = 0.768
0.768     * 2 = 1.563
0.536     * 2 = 1.072
...           ...
```

$$0.5685_{dec} \approx 0.100100011_{bin}(0.568359375)$$

Dvojková soustava

- Přepočet z desítkové do dvojkové soustavy:

$$168.5685_{dec} \approx 10101000.100100011_{bin}$$

- Nemá ukončený "desetinný"(dvojkový) rozvoj, protože:

$$168.5685 = 168 + \frac{1137}{2000}$$

  a jmenovatel 2000 není mocnina 2.

- Pozor:

$$0.1_{dec} = 0.0001100110011\ldots$$

Dvojková soustava

Převeďte 11010.01101_{bin} do desítkové soustavy:

Dvojková soustava

Převeďte 11010.01101_{bin} do desítkové soustavy:

```
1 *  0.03125      =  0.03125
0 *  0.0625       =  0
1 *  0.125        =  0.125
1 *  0.25         =  0.25
0 *  0.5          =  0
0 *  1            =  0
1 *  2            =  2
0 *  4            =  0
1 *  8            =  8
1 * 16            = 16
sum                  26.40625
```

$$11010.01101_{bin} = 26.40625_{dec}$$

Celá čísla se znaménkem

- Kódování:
  - doplňkový kód (dvojkový doplněk),
  - inverzní kód,
  - aditivní kód,
  - přímý kód.
- Typické použití:
  - doplňkový kód – celá čísla,
  - aditivní kód – exponent desetinných čísel,
  - přímý kód – mantisa desetinných čísel.

Doplňkový kód

- Pro n bitovou reprezentaci je rozsah -2^{n-1} až $2^{n-1} - 1$.
- Kladná čísla a 0 se ukládají beze změny.
- Záporné číslo x je uloženo jako hodnota $2^n + x$ $(= \bar{x} + 1)$. Tedy:
  - kladná čísla a nula mají v nejvyšším bitu 0,
  - záporná čísla mají v nejvyšším bitu 1.
- Snadné sčítání, odčítání a porovnávání.
- Obtížnější (ale zvládnutelné) násobení a dělení.

Přímý kód

- Pro n bitovou reprezentaci je rozsah $-2^{n-1}+1$ až $2^{n-1}-1$.
- Nižších $n-1$ bitů ukládá absolutní hodnotu čísla.
- Nejvyšší bit kóduje znaménko (0 kladné, 1 záporné).
- Existují dvě různé nuly: $+0$ a -0.
- Snadné násobení a dělení.
- Pracnější sčítání a odčítání.

Aditivní kód

- Čísla se ukládají zvětšená o konstantu A.
- Typicky $A = 2^{n-1}$ nebo $A = 2^{n-1} - 1$.
- Nejvyšší bit určuje znaménko (0 záporné, 1 kladné).
- Snadné sčítání, odčítání a porovnávání.
- Nevhodný pro násobení a dělení.

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaká je reprezentace:

	doplňkový kód	přímý kód
108		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaká je reprezentace:

	doplňkový kód	přímý kód
108	0110 1100	0110 1100
-108		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaká je reprezentace:

	doplňkový kód	přímý kód
108	0110 1100	0110 1100
-108	1001 0100	1110 1100
36		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaká je reprezentace:

	doplňkový kód	přímý kód
108	0110 1100	0110 1100
-108	1001 0100	1110 1100
36	0010 0100	0010 0100
-36		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaká je reprezentace:

	doplňkový kód	přímý kód
108	0110 1100	0110 1100
-108	1001 0100	1110 1100
36	0010 0100	0010 0100
-36	1101 1100	1010 0100
0		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaká je reprezentace:

	doplňkový kód	přímý kód
108	0110 1100	0110 1100
-108	1001 0100	1110 1100
36	0010 0100	0010 0100
-36	1101 1100	1010 0100
0	0000 0000	0000 0000
		1000 0000

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010	90	90
1101 1110		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010	90	90
1101 1110	-34	-94
1100 0001		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010	90	90
1101 1110	-34	-94
1100 0001	-63	-65
1010 1010		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010	90	90
1101 1110	-34	-94
1100 0001	-63	-65
1010 1010	-86	-42
1111 1111		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010	90	90
1101 1110	-34	-94
1100 0001	-63	-65
1010 1010	-86	-42
1111 1111	-1	-127
1000 0000		

Čísla se znaménkem

Mějme čísla kódovaná v $n = 8$ bitech. Jaké číslo uchovává reprezentace:

	doplňkový kód	přímý kód
0101 1010	90	90
1101 1110	-34	-94
1100 0001	-63	-65
1010 1010	-86	-42
1111 1111	-1	-127
1000 0000	-128	-0

Desetinná čísla

- základ b, typicky $b = 2$,
- mantisa m fixní velikosti, $\frac{1}{b} \leq |m| < 1$,
- exponent e fixní velikosti,
- znaménko s (1 bit).

$$x = (-1)^s \times m \times b^e = (-1)^s \times m \times 2^e$$

Doporučení IEEE 754:

velikost	mantisa	exponent	rozsah
4 B	23 + 1b	8b	$\approx 1.18 \times 10^{-38} .. \approx 3.4 \times 10^{38}$
8 B	52 + 1b	11b	$\approx 2.23 \times 10^{-308} .. \approx 1.8 \times 10^{308}$
10 B	64 + 1b	15b	$\approx 3.36 \times 10^{-4932} .. \approx 1.19 \times 10^{4932}$

Desetinná čísla

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

Desetinná čísla

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Převedeme celou část:

$$123_{10} = 1111011_2$$

Desetinná čísla

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Převedeme celou část:

$$123_{10} = 1111011_2$$

- Převedeme desetinnou část:

$$0.53125_{10} = 0.10001_2$$

Desetinná čísla

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Převedeme celou část:

$$123_{10} = 1111011_2$$

- Převedeme desetinnou část:

$$0.53125_{10} = 0.10001_2$$

- Určíme mantisu:

$$m = 1111011.10001_2$$

Desetinná čísla

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Převedeme celou část:

$$123_{10} = 1111011_2$$

- Převedeme desetinnou část:

$$0.53125_{10} = 0.10001_2$$

- Určíme mantisu:

$$m = 1111011.10001_2$$

- Normalizace mantisy:

$$m = 1.11101110001_2 * 2^6$$

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Normalizace mantisy:

$$m = 1.11101110001_2 * 2^6$$

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Normalizace mantisy:

$$m = 1.11101110001_2 * 2^6$$

- Odstranění skryté jedničky, doplnění na 23 bitů:

$$11101110001000000000000_2$$

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Normalizace mantisy:

$$m = 1.11101110001_2 * 2^6$$

- Odstranění skryté jedničky, doplnění na 23 bitů:

$$11101110001000000000000_2$$

- Určení exponentu (v aditivním kódu, 8 bitů):

$$(6 + 127)_{10} = 10000101_2$$

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Normalizace mantisy:

$$m = 1.11101110001_2 * 2^6$$

- Odstranění skryté jedničky, doplnění na 23 bitů:

$$11101110001000000000000_2$$

- Určení exponentu (v aditivním kódu, 8 bitů):

$$(6 + 127)_{10} = 10000101_2$$

- Kladné číslo – znaménko

$$s = 0$$

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Výsledek:

  `znaménko exponent mantisa`

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Výsledek:

  ```
  znaménko exponent mantisa

  = 0  10000101  11101110001000000000000 (bin)
  ```

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Výsledek:

  `znaménko exponent mantisa`

  `= 0  10000101  11101110001000000000000 (bin)`

  `= 01000010  11110111  00010000  00000000 (bin)`

Desetinná čísla (pokračování)

Převod čísla 123.53125 do IEEE reprezentace (32 bitů):

- Výsledek:

```
znaménko exponent mantisa

 = 0  10000101  11101110001000000000000 (bin)

 = 01000010  11110111  00010000  00000000 (bin)

 = 42  f7  10  00 (hex)
```

Desetinná čísla - aritmetické operace

Jak se sečte $20.5 + 14.5$ (IEEE, 32 bit):

Desetinná čísla - aritmetické operace

Jak se sečte 20.5 + 14.5 (IEEE, 32 bit):

```
20.5 (dec) = 10100.1 (bin)
  = 1.01001000000000000000000 * 2^4
14.5 (dec) = 1110.1 (bin)
  = 1.01010000000000000000000 * 2^3
```

Desetinná čísla - aritmetické operace

Jak se sečte 20.5 + 14.5 (IEEE, 32 bit):

```
20.5 (dec) = 10100.1 (bin)
  = 1.01001000000000000000000 * 2^4
14.5 (dec) = 1110.1 (bin)
  = 1.01010000000000000000000 * 2^3
```

Převod na společný exponent (zde 4):

```
14.5 (dec) = 1110.1 (bin)
  = 0.11101000000000000000000 * 2^4
```

Desetinná čísla - aritmetické operace

Jak se sečte 20.5 + 14.5 (IEEE, 32 bit):

```
20.5 (dec) = 10100.1 (bin)
  = 1.01001000000000000000000 * 2^4
14.5 (dec) = 1110.1 (bin)
  = 1.01010000000000000000000 * 2^3
```

Převod na společný exponent (zde 4):

```
14.5 (dec) = 1110.1 (bin)
  = 0.11101000000000000000000 * 2^4
```

Součet mantis a případná normalizace:

```
    1.01001000000000000000000 * 2^4
  + 0.11101000000000000000000 * 2^4
  =10.00110000000000000000000 * 2^4
```

Desetinná čísla - aritmetické operace

Jak se sečte 20.5 + 14.5 (IEEE, 32 bit):

```
20.5 (dec) = 10100.1 (bin)
  = 1.01001000000000000000000 * 2^4
14.5 (dec) = 1110.1 (bin)
  = 1.01010000000000000000000 * 2^3
```

Převod na společný exponent (zde 4):

```
14.5 (dec) = 1110.1 (bin)
  = 0.11101000000000000000000 * 2^4
```

Součet mantis a případná normalizace:

```
    1.01001000000000000000000 * 2^4
  + 0.11101000000000000000000 * 2^4
  =10.00110000000000000000000 * 2^4
  = 1.00011000000000000000000 * 2^5
```

Desetinná čísla - aritmetické operace

Jak se sečte 20.5 + 14.5 (IEEE, 32 bit):

```
20.5 (dec) = 10100.1 (bin)
  = 1.01001000000000000000000 * 2^4
14.5 (dec) = 1110.1 (bin)
  = 1.01010000000000000000000 * 2^3
```

Převod na společný exponent (zde 4):

```
14.5 (dec) = 1110.1 (bin)
  = 0.11101000000000000000000 * 2^4
```

Součet mantis a případná normalizace:

```
    1.01001000000000000000000 * 2^4
+ 0.11101000000000000000000 * 2^4
=10.00110000000000000000000 * 2^4
= 1.00011000000000000000000 * 2^5
= 100011.0 (bin) = 35 (dec)
```

Desetinná čísla - aritmetické operace

Jak se sečte 20000000 + 9 (IEEE, 32 bit):

Desetinná čísla - aritmetické operace

Jak se sečte 20000000 + 9 (IEEE, 32 bit):

```
20000000 (dec) = 10011000100101101000000000.0 (bin)
  = 1.00110001001011010000000 * 2^24
9 (dec) = 1001.0 (bin)
  = 1.00100000000000000000000 * 2^3
```

Desetinná čísla - aritmetické operace

Jak se sečte 20000000 + 9 (IEEE, 32 bit):

```
20000000 (dec) = 10011000100101101000000000.0 (bin)
  = 1.00110001001011010000000 * 2^24
9 (dec) = 1001.0 (bin)
  = 1.00100000000000000000000 * 2^3
```

Převod na společný exponent (zde 24):

```
9 (dec) = 1001.0 (bin)
  = 0.000000000000000000001001 * 2^24
```

Desetinná čísla - aritmetické operace

Jak se sečte 20000000 + 9 (IEEE, 32 bit):

```
20000000 (dec) = 10011000100101101000000000.0 (bin)
  = 1.00110001001011010000000 * 2^24
9 (dec) = 1001.0 (bin)
  = 1.00100000000000000000000 * 2^3
```

Převod na společný exponent (zde 24):

```
9 (dec) = 1001.0 (bin)
  = 0.00000000000000000000100 * 2^24
```

Společný exponent zde vede ke ztrátě přesnosti

Desetinná čísla - aritmetické operace

Jak se sečte 20000000 + 9 (IEEE, 32 bit):

```
20000000 (dec) = 10011000100101101000000000.0 (bin)
  = 1.00110001001011010000000 * 2^24
9 (dec) = 1001.0 (bin)
  = 1.00100000000000000000000 * 2^3
```

Převod na společný exponent (zde 24):

```
9 (dec) = 1001.0 (bin)
  = 0.00000000000000000000100 * 2^24
```

Společný exponent zde vede ke ztrátě přesnosti

Součet mantis a případná normalizace:

```
    1.00110001001011010000000 * 2^24
+ 0.00000000000000000000100 * 2^24
= 1.00110001001011010000100 * 2^24
```

Desetinná čísla - aritmetické operace

Jak se sečte 20000000 + 9 (IEEE, 32 bit):

```
20000000 (dec) = 10011000100101101000000000.0 (bin)
  = 1.00110001001011010000000 * 2^24
9 (dec) = 1001.0 (bin)
  = 1.00100000000000000000000 * 2^3
```

Převod na společný exponent (zde 24):

```
9 (dec) = 1001.0 (bin)
  = 0.00000000000000000000100 * 2^24
```

Společný exponent zde vede ke ztrátě přesnosti

Součet mantis a případná normalizace:

```
    1.00110001001011010000000 * 2^24
+ 0.00000000000000000000100 * 2^24
= 1.00110001001011010000100 * 2^24
= 20000008 (dec)
```

Desetinná čísla

- Jak se určí počet platných cifer?

Desetinná čísla

- Jak se určí počet platných cifer?

$$x = \frac{\text{mantisa_bitů}}{\log_2 10}$$

- Jak se určí rozsah?

Desetinná čísla

- Jak se určí počet platných cifer?

$$x = \frac{\text{mantisa_bitů}}{\log_2 10}$$

- Jak se určí rozsah?

$$x = 2^{2^{\text{exponent_bitů}} - 1}$$

- Nejmenší a největší hodnoty exponentu jsou rezervované pro speciální hodnoty (inf, NaN, 0), skutečný rozsah je tedy poloviční.

Doporučení IEEE 754:

velikost	mantisa	platné des. cifry
4 B	23 + skrytá 1	7.22
8 B	52 + skrytá 1	15.95
10 B	64	19.26

Znaky, řetězce

- Znaky kódované čísly:
  - každý znak má přiřazené nějaké číslo,
  - přiřazení určené tabulkou (norma, dohoda).
- ASCII
  - 128 pozic, znaky US abacedy, číslice, speciální znaky,
  - chybí znaky používané v jiných jazycích (slovanské, řecké, azbuka, čínské, ...).
- Rozšířená ASCII:
  - Prvních 128 pozic = ASCII, nad 128 znaky národních abeced,
  - různé tabulky pro různé státy (iso 8859-2, cp1250, ...).
- UNICODE:
  - všechny (většina) používaných znaků,
  - velikost tabulky = $2^{20} + 2^{16}$ pozic,
  - prvních 128 pozic = ASCII.

Znaky, řetězce

Kódování znaků v paměti počítače:

- ASCII (rozšířená ASCII):
  - 1 bajt = znak.
- UNICODE:
  - 2 bajty = znak (UCS-2),
  - 4 bajty = znak (UCS-4 = UTF-32),
  - 2 nebo 4 bajty = znak (UTF-16),
  - 1-4 bajty = znak (UTF-8).

Znaky, řetězce

Příklad kódování řetězce `Žlutý kůň`:

- cp1250:

  ```
  8e 6c 75 74 fd 20 6b f9 f2
  ```

- iso-8859-2:

  ```
  ae 6c 75 74 fd 20 6b f9 f2
  ```

- UTF-32:

  ```
  7d 01 00 00 6c 00 00 00 75 00 00 00 74 00 00 00
  fd 00 00 00 20 00 00 00 6b 00 00 00 6f 01 00 00
  48 01 00 00
  ```

- UTF-16:

  ```
  7d 01 6c 00 75 00 74 00 fd 00 20 00 6b 00 6f 01
  48 01
  ```

- UTF-8:

  ```
  c5 bd 6c 75 74 c3 bd 20 6b c5 af c5 88
  ```

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

- UTF-8 řetězec:

  ```
  ČVUT...
  ```

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

- UTF-8 řetězec:

  `ČVUT...`
- Windows-1250 řetězec:

  `ÄŚVUT...`

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

- UTF-8 řetězec:

  `ČVUT...`
- Windows-1250 řetězec:

  `ÄŚVUT...`
- Desetinné číslo, 64b, big-endian:

  `-1.6727353762607028764672e22`

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

- UTF-8 řetězec:

  `ČVUT...`
- Windows-1250 řetězec:

  `ÄŚVUT...`
- Desetinné číslo, 64b, big-endian:

  `-1.6727353762607028764672e22`
- Desetinné číslo, 64b, little-endian:

  `3.0343418641270625830483542883 3e-86`

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

- UTF-8 řetězec:

  `ČVUT...`
- Windows-1250 řetězec:

  `ÄŚVUT...`
- Desetinné číslo, 64b, big-endian:

  `-1.6727353762607028764672e22`
- Desetinné číslo, 64b, little-endian:

  `3.0343418641270625830483542883 3e-86`
- Celé číslo bez znaménka, 64b, little-endian:

  `3327648114417241284`

Znaky, řetězce

Co znamená následující paměťový obsah (bajty):

```
c4 8c 56 55 54 2e 2e 2e
```

- UTF-8 řetězec:

  `ČVUT...`
- Windows-1250 řetězec:

  `ÄŚVUT...`
- Desetinné číslo, 64b, big-endian:

  `-1.6727353762607028764672e22`
- Desetinné číslo, 64b, little-endian:

  `3.0343418641270625830483542883 3e-86`
- Celé číslo bez znaménka, 64b, little-endian:

  `3327648114417241284`
- ...

Dotazy . . .

Děkuji za pozornost.