ISC Cvičení 2 - Čísla

Jakub Martiško

4. října 2016

Polynomiální zápis

- Používáme poziční soustavy — umístění číslice v rámci čísla je důležité (123 vs. 321).
- Předpokládejme desítkovou soustavu:

$$1\,323.52 = 1*10^3+3*10^2+2*10^1+3*10^0+5*10^{-1}+2*10^{-2}$$

- Lze použít pro převod z libovolné soustavy do desítkové:

$$(101.1)_2 = (1*2^2+0*2^1+1*2^0+1*2^{-1})_{10} = (5.5)_{10}$$

Převod mezi soustavami - celá část

function CONVERT(*number*, *base*)
    result $\leftarrow 0$
    remainder $\leftarrow 0$
    while (*number* > 0) **do**
        remainder $\leftarrow$ *number*%*base* $\triangleright$ 5%2=1
        number $\leftarrow$ *number*/*base* $\triangleright$ 5/2=2
        result $\leftarrow$ *remainder* $++$ *result* $\triangleright$ 12++34 = 1234
    return *result*

Převod mezi soustavami - desetiná část

function CONVERT($number, base$)
    $result \leftarrow 0$
    $remainder \leftarrow 0$
    while $(number! = 0)$ **do**
        $aux = number * base$
        $remainder \leftarrow trunc(aux)$ ▷ $trunc(3.9) = 3$
        $number \leftarrow aux - remainder$
        $result \leftarrow result + +remainder$ ▷ $12++34 = 1234$
    return $result$

Sčítání ve dvojkové soustavě

- Podobné jako v desítkové

$$0+0=0; 0+1=1; 1+1=10; 1+1+1=11; \ldots$$

- Např:

$$\begin{array}{cccc} & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 1 & 0 \\ \hline & 1 & 0 & 1 & 1 \end{array}$$

Záporná čísla

- Přímý kód — nejvyšší bit značí znaménko $(0 \approx +, 1 \approx -)$

$$(0000\,1001)_2 = (+5)_{10}$$

$$(1000\,1001)_2 = (-5)_{10}$$

- Inverzní kód (jedničkový doplněk)— převrácené hodnoty jednotlivých bitů, nejvyšší bit stále určuje znaménko

$$(0000\,1001)_2 = (+5)_{10}$$

$$(1111\,0110)_2 = (-5)_{10}$$

Záporná čísla - doplňkový kód

- Doplňkový kód (dvojkový doplněk) — vezmu inverzní kód a přičtu jedničku:

$$(6)_{10} = (0000\,0110)_2 \rightarrow (1111\,1001)_2 \rightarrow (1111\,1010)_2 = (-6)_{10}$$

- Nebo — jdu od nejméně významného bitu, nuly opisuju, jakmile narazím na jedničku tak ji také opíšu a další čísla invertuju:

$$(0000\,1010) \rightarrow (0) \rightarrow (10) \rightarrow (110) \rightarrow \cdots \rightarrow (1111\,0110)_2$$

Převody mezi soustavami o základu 2, 8, 16, 32 ...

- Použít algoritmus ze strany 4
- Jedná se o soustavy o základu 2^n:

2	16(8)	2	16(8)	2	16	2	16
0000	0	0100	4	1000	8	1100	C
0001	1	0101	5	1001	9	1101	D
0010	2	0110	6	1010	A	1110	E
0011	3	0111	7	1011	B	1111	F

Endiany

- Určují pořadí jednotlivých bajtů, při ukládání dlouhých hodnot do paměti
- Big: některé síťové protokoly
- Little: x86, x86-64
- Např pro číslo *FFEEDDCC*:

adresa	Big	Little
100	FF	CC
101	EE	DD
102	DD	EE
103	CC	FF

Záporná čísla - transformovaná (posunutá) nula

- Nula není reprezentována jako $\ldots 0000$ ale je posunutá - většinou do hodnoty, kde $MSB = 1$ a ostatní bity jsou rovny 0
- Číslo obsahující samé 0 pak odpovídá nejmenšímu číslu a číslo obsahující samé 1 pak největšímu číslu.
- Převod — inverze MSB v doplňkovém kódu (platí jen při posunu definovaném výše).
- Např. (8 bitů):

$$(1000\,0000)_2 = (0)_{10}; (3)_{10} = (1000\,0011)_2; (-1)_{10} = (0111\,1111)_2$$

Plovoucí des. čárka

- Číslo se rozdělí na znaménko, mantisu a exponent, výsledek je pak dán vztahem:

$$vysl = mantisa * zaklad^{exponent}$$

- Základ je roven hodnotě 2 (dvojková soustava), mantisa v přímém kódu, exponent v posunuté nule (může být posunuta jinak než bylo popsáno výše).
- Např. (pro jednoduchost uvedeno v desítkové soustavě):

$$-1.2345 = -(12345 * 10^{-4}) = [-][12345][-4]$$