Zobrazování celých čísel v počítači

Bit, byte, slovo

Jednotkou informace je jeden **bit** *(**B**inary digi**T**).* Bit může nabývat dvou hodnot: jedničky a nuly.

Skupina osmi bitů se označuje jako **byte** (též bajt nebo slabika). Byte může nabývat $2^8 = 256$ stavů. Může tedy nabývat hodnot 0 až 255 v soustavě desítkové či 0 až FF v soustavě šestnáctkové.

Větší jednotky než byte mají standardní velikosti, jsou to mocniny dvou, názvy jsou však v literatuře nejednoznačné.

Dva bajty se někdy označují jako **slovo poloviční délky** (půlslovo – *Half Word*). Půlslovo může nabývat $2^{16} = 65\,536$ stavů, což představuje desítkové čísla 0 až 65 535 či šestnáctková čísla 0 až FFFF.

V závislosti na **architektuře uspořádání bajtů** hovoříme o:

- **little endian** – kdy v paměti je uložen nejprve méně významný bajt **LSB** *(Less Signicant Byte*), následovaný více významným bajtem **MSB** *(Most Signicant Byte).* Tuto architekturu používá např. firma Intel
- **big endian** – kdy v paměti je uložen nejdříve MSB a pak teprve LSB. Tento způsob používají např. procesory firmy Motorola.

Čtyři bajty představují **slovo** *(Word),* tedy dvě půlslova. Slovo může nabýt $2^{32} = 4\,294\,967\,296$ stavů.

Osm bajtů je **slovo dvojité délky** (dvojslovo – *Double Word*).

Číselné soustavy

Číselné soustavy lze rozdělit na:

- **nepoziční** – hodnota číslice nezáleží na pozici v čísle. Takovou soustavou je např. soustava římských číslic
- **poziční** – hodnota číslice závisí na pozici v čísle. Příkladem je desítková soustava.

Zapíšeme-li číslo $1234_{(10)}$ znamená to:

$$1234 = 1 \,.\, 10^3 + 2 \,.\, 10^2 + 3 \,.\, 10^1 + 4 \,.\, 10^0$$

Dolní index u čísla $1234_{(10)}$ představuje základ soustavy – deset, takže zápis čísla je v desítkové soustavě.

Dvojková soustava

Podobně lze zapsat číslo $10111_{(2)}$:

$$10111_{(2)} = 1 \,.\, 2^4 + 0 \,.\, 2^3 + 1 \,.\, 2^2 + 1 \,.\, 2^1 + 1 \,.\, 2^0$$

Dolní index u čísla (2) znamená, že toto číslo je v soustavě **dvojkové** *(binární).* Dvojková číslice má jen dvě číslice 0 a 1.

Chceme-li číslo $10111_{(2)}$ vyjádřit v soustavě desítkové, musíme vypočítat výše uvedený rozvoj.

$$10111_{(2)} = 1\ .\ 2^4 + 0\ .\ 2^3 + 1\ .\ 2^2 + 1\ .\ 2^1 + 1\ .\ 2^0 = 1\ .\ 16 + 0\ .\ 8 + 1\ .\ 4 + 1\ .\ 2 + 1\ .\ 1 = 23_{(10)}$$

Převod z desítkového čísla na číslo dvojkové lze provádět postupným dělením desítkového čísla dvojkou s vyjadřováním zbytků jako např. při převodu čísla $27_{(10)}$ do dvojkové soustavy:

27 : 2 = 13	1
13 : 2 = 6	1
6 : 2 = 3	0
3 : 2 = 1	1
1 : 2 = 0	1

Zbytky po dělení se zapíší v opačném pořadí, než v jakém byly získány po dělení, tedy zdola nahoru. Zápis zbytků pak představuje výsledek, který je $11011_{(2)}$.

Zápis dvojkového čísla je pro člověka nepřehledný, protože číslo v dvojkové soustavě je v průměru 3,3krát delší než odpovídající zápis čísla v desítkové soustavě. Pro zjednodušení zápisu ve dvojkové soustavě se proto ve výpočetní technice užívá pomocných soustav – osmičkové a šestnáctkové.

Osmičková soustava

V **osmičkové (oktalové) soustavě** se využívá toho, že $2^3 = 8$. Dvojkové číslo se proto zprava rozdělí do trojic dvojkových číslic a zleva se případně doplní nulami, pokud není poslední trojice úplná. Takto upravené dvojkové číslo se převede na osmičkové tím, že se každá trojice nul a jedniček vyjádří jako odpovídající osmičková číslice. Osmičková soustava obsahuje čísla od 0 do 7.

Příklad: $27_{(10)} = 11011_{(2)}$. Jak vyjádříme toto číslo v osmičkové soustavě?

Rozdělíme číslo 11011 na trojice zprava 011 011. Trojice dvojkových číslic nahradíme příslušnými osmičkovými číslicemi tedy 33. Máme výsledek:

$$27_{(10)} = 11011_{(2)} = 33_{(8)}$$

Číslo $33_{(8)}$ je $3\ .\ 8^1 + 3\ .\ 8^0 = 3\ .\ 8 + 3\ .\ 1 = 27_{(10)}$

Šestnáctkové soustava

V **šestnáctkové (hexadecimální) soustavě** se využívá toho, že $2^4 = 16$. Dvojkové číslo se proto zprava rozdělí do čtveřic dvojkových číslic a zleva se případně doplní nulami, pokud není poslední čtveřice úplná. Takto upravené dvojkové číslo se převede na šestnáctkové tím, že se každá čtveřice nul a jedniček vyjádří jako odpovídající šestnáctková číslice. Pro šestnáctkovou soustavu potřebujeme 16 číslic, avšak desítková jich nabízí jen 10, takže použijeme ještě číslice A až F pro vyjádření desítky až patnáctky.

Příklad: $27_{(10)} = 11011_{(2)}$. Jak vyjádříme toto číslo v šestnáctkové soustavě?

Rozdělíme číslo 11011 na čtveřice zprava 0001 1011. Čtveřice dvojkových číslic nahradíme příslušnými šestnáctkovými číslicemi tedy 1B. Máme výsledek:

$$27_{(10)} = 11011_{(2)} = 33_{(8)} = 1B_{(16)}$$

Číslo $1B_{(16)}$ je $1\ .\ 16^1 + 11\ .\ 16^0 = 1\ .\ 16 + 11\ .\ 1 = 27_{(10)}$

Převod z desítkové soustavy do libovolné soustavy

Nechť je C celé číslo v desítkové soustavě. Převod do soustavy se základem Z se provede následujícím způsobem:

$$C : Z = C_0 \text{ zbytek } a_0$$
$$C_0 : Z = C_1 \text{ zbytek } a_1$$
$$\ldots \quad \ldots$$
$$C_{n-1} : Z = 0 \text{ zbytek } a_n$$

Převedené číslo je $a_n\, a_{n-1} \ldots a_1\, a_{0\ (Z)}$

Záporná celá čísla

V bajtu, slově apod. mohou být jednak čísla bez znaménka *(unsigned)* a jednak čísla se znaménkem *(signed).* Pro znaménko se užívá nejvyšší bit čísla. Pro vyjádření záporných čísel se užívá těchto kódů:

- **Přímého** – znaménkový bit obsahuje jedničku a ostatní bity čísla jsou beze změny. Rozdělení na kladná a záporná čísla je symetrické, nula má dva obrazy +0 a -0.
- **Inverzního** – záporné číslo dostaneme z kladného překlopením nul na jedničky a naopak. Rozdělení na kladná a záporná čísla je symetrické, nula má dva obrazy +0 a -0.
- **Dvojkového doplňku** – je to inverze +1. Rozdělení na kladná a záporná čísla je nesymetrické – rozsah záporných čísel je o 1 větší; nula má jen jeden obraz. Tento kód je výhodný pro výpočty, neboť při odčítání dvou čísel se menšitel převede do doplňkového kódu a rozdíl získáme sečtením menšence a menšitele převedeného do doplňkového kódu.

V následující tabulce bude vše ilustrováno. Pro jednoduchost předpokládejme, že pro zobrazování čísel se používá čtveřic bitů:

Obsah (bity)	Bez znaménka (desítkově)	Se znaménkem (desítkově) způsob		
		přímý	inverzní	doplňkový
0000	0	0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	-0	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	-0	-1

Tab. 1 – Kódy záporných čísel

Literatura:

[1] Rubeš, J.: Nebojte se programovat, Computer Media, Bedihošť 2001

[2] Krejčí, A: Technické vybavení počítačů, text přednášky z www.fce.vutbr.cz

[3] Blatný, J. a kol.: Číslicové počítače, SNTL, Praha 1982