

Zobrazení dat v počítači – číselné soustavy, převody (včetně desetinných čísel), čísla v pohyblivé čárce

Data a informace

Nejmenší jednotkou velikosti dat je bit (b). Ten byl zvolen pro snadnější reprezentaci čísel a znaků v počítačích – stačí pouze dvě úrovně napětí elektrického signálu – hodnoty 0 a 1. Jednotkou kapacity paměti je pak byte (B). $1 \text{ B} = 8 \text{ b}$.

V momentě, kdy přiřadíme datům význam, se data stanou informacemi. Když pak informacím přiřadíme také kontext, stanou se z nich znalosti.

Bajt pak má kromě ostatních bitů nejvýznamnější bit (MSB) (7) a nejméně významný bit (LSB) (0).

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Nejmenší číslo zapsané v 1 bajtu je 00000000 (0) a největší 11111111 (255). 1 bajt lze tedy přivést do 256 různých stavů.

Číselné soustavy obecně

Číselná soustava je způsob prezentace čísel – pro různé druhy uložení nebo využití mohou být vhodnější různé druhy soustav. V registrech jsou však veškerá data uložena v soustavě binární.

Převod mezi soustavami o různém základu se běžně praktikuje přes desítkovou soustavu – vyjma soustav, kde základ jedné soustavy je kladnou celočíselnou mocninou soustavy druhé, kde lze postupovat i přímo – *příklad uveden níže u binární soustavy (*)*.

Decimální/dekadická/desítková (dec)

V desítkové soustavě počítají lidé běžně. Pro zápis čísel se využívají znaky 0 až 9.

Převod celých čísel do binární soustavy

Z decimální do binární soustavy se číslo nejjednodušeji převádí postupným dělením dvěma.

1. $55_{\text{D}} \div 2_{\text{D}} = 27_{\text{D}}$ (zbytek 1)
2. $27_{\text{D}} \div 2_{\text{D}} = 13_{\text{D}}$ (zbytek 1)
3. $13_{\text{D}} \div 2_{\text{D}} = 6_{\text{D}}$ (zbytek 1)
4. $6_{\text{D}} \div 2_{\text{D}} = 3_{\text{D}}$ (zbytek 0)
5. $3_{\text{D}} \div 2_{\text{D}} = 1_{\text{D}}$ (zbytek 1)
6. $1_{\text{D}} \div 2_{\text{D}} = 0_{\text{D}}$ (zbytek 1)

↑
|
|
|
|
|
|

V momentě, kdy nám po dělení vyjde 0_{D} , zapíšeme poslední zbytek, počínaje jímž poté všechny zbytky sepíšeme k sobě). Převedené číslo tedy vzniká ze zbytků po dělení.

55_{D} se tedy rovná 110111_{B} .

Převod desetinných čísel do binární soustavy

Z decimální do binární soustavy se desetinné číslo nejjednodušeji převádí postupným násobením desetinné části dvěma.

- | | | | | |
|----|-----------------------------|----------------------|---|--|
| 1. | $0,125_D \times 2_D = 0,25$ | (zbytek po dělení 0) | ↑ | V momentě, kdy nám vyjde nulový výsledek v desetinné části postupně sepíšeme zbytky po dělení. |
| 2. | $0,25_D \times 2_D = 0,50$ | (zbytek po dělení 0) | | |
| 3. | $0,50_D \times 2_D = 1,00$ | (zbytek po dělení 1) | | |

Binární/dvojková (bin)

$0,125_D$ se tedy rovná $0,001_B$.

Pro zápis čísel využívá dva znaky: 0 a 1. Kromě toho, že v této soustavě jsou zapsána veškerá data v registrech, využívají binární soustavu také logické obvody, kde lze stavy zapsat pouze dvěma způsoby (např. 0 = nepravda = proud neprotéká; 1 = pravda = proud protéká).

dec. (10)	bin. (2)
1	1
10	1010
55	11 0111
128	1000 0000

Registry mají vždy určitou šířku a číslo je tedy nutné zapsat s takovým počtem cifer, jako má daný registr (např. u 16b procesorů mají registry délku 16 nebo 32 bitů).

(*) U soustav se základem 2^n , kde $n > 1$, je snadný převod do jiné takové soustavy (plus do decimální) právě přes dvojkovou soustavu rozdělením čísla na jednotlivé číslice, vyjádřením každé číslice ve dvojkové soustavě a následně

- u převodu do desítkové soustavy sečtením hodnot jednotlivých cifer;
- u převodu do jiné soustavy o základu 2^n , kde $n > 1$, rozdělením celého řetězce nul a jedniček na části po n číslicích, sečtením hodnot jednotlivých cifer v každé části a spojením součtů do jednoho číselného řetězce.

U binární soustavy lze dále dělit způsoby zápisu na několik druhů, každý pro jiné případy:

- přímý kód** – používá se jen u přirozených čísel
- inverzní kód** (jednotkový doplněk) – nejvyšší bit je znaménko, v počítačích se běžně nepoužívá; kladné číslo začíná 0 a zbylé řády jsou vyjádřeny přímým kódem; záporné číslo je vytvořeno bitovou inverzí absolutní hodnoty čísla a nejvyšší bit má hodnotu 1
- dvojkový doplněk** (doplňkový kód) – kladné celé číslo je v doplňkovém kódu zobrazeno přímo, ale nejvyšší bit má hodnotu 0, u záporného čísla je nutné provést bitovou inverzi a přičíst 1; ke dvojkovému doplňku lze provést dvojkový doplněk (= k zaznamenanému číslu číslo opačné)
- float** – čísla s desetinnou čárkou – skládají se z *mantisy* a *exponentu*; u dvojkové soustavy je exponent posunutí řádu (s posunutou nulou) a mantisa normované číslo; např. podle normy IEEE 754 single (šířka 32 bitů) *mantisa* zabírá 23+1 bitů (1 navíc pro znaménko) a 8 bitů pro *exponent*

Oktalová/oktální/osmičková (oct)

Pro zápis čísel se využívají znaky 0 až 7. Používá se například v OS Linux pro nastavení výše oprávnění pro majitele souboru, skupinu a ostatní. K označení, že jde o oktalovou soustavu, se využívá číslice 0 předcházející samotné hodnotě (místo 64 tedy např. 064).

dec. (10)	oct. (8)
1	1
10	12
55	67
128	200

Hexadecimální/šestnáctková (hex)

Pro zápis čísel se využívají znaky 0 až 9 a A až F. Soustava se používá mj. např. pro adresy v operační paměti počítače. K označení, že jde o hexadecimální soustavu, se používá nejčastěji zápisu s předponou 0x (místo 37 tedy např. 0x37).

Hornerovo schéma

Pomocí Hornerova schématu lze snadno převést číslo z libovolné soustavy do soustavy desítkové. Běžně se pro takový převod používá. Je dokázáno¹, že jde o nejefektivnější algoritmus (v oboru \mathbb{R}).

dec. (10)	hex. (16)
1	1
10	A
55	37
128	80

$$a_n \times Z^n + a_{n-1} \times Z^{n-1} + \dots + a_1 \times Z^1 + a_0 \times Z^0 + a_{-1} \times Z^{-1} + \dots + a_{-m} \times Z^{-m}$$

a = koeficient číselné soustavy – číslice v dané soustavě; Z = základ číselné soustavy

Příklad pro binární soustavu

$$a = 0; 1$$

$$Z = 2$$

$$10011_B = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 16 + 2 + 1 = 19_D$$

Příklad pro osmičkovou soustavu

$$a = 0; 1; 2; 3; 4; 5; 6; 7$$

$$Z = 8$$

$$146_O = 1 \times 8^2 + 4 \times 8^1 + 1 \times 8^0 = 1 \times 64 + 4 \times 8 + 6 \times 1 = 64 + 32 + 6 = 102_D$$

Kódy

Kód je předpis, podle kterého se data převádějí do jiné reprezentace za účelem jejich přenosu či záznamu. V případě číslicové techniky na bity. Jiným známým způsobem kódování je převod písmen a číslic do Morseovy abecedy.

Kódové slovo je bitová reprezentace daného znaku, *kódování* je proces převodu dat na symboly kódu, *dekódování* je opačný proces, *šifrování* je pak takové kódování, kde algoritmus převodu není běžně dostupný.

Detekční a opravné kódy

Detekční kódy umožňují detekci chyb. Základní detekční kódy detekují pouze jednu chybu. Nejčastější a nejjednodušší je *parita* – přidání 9. bit doplňující součet zprávy/bajtu na **a**) sudý počet jedniček (sudá – even – parita) a **b**) lichý počet jedniček (lichá – odd – parita).

Opravné kódy opravují zjištěnou chybu. Lze použít křížovou paritu pro odhalení jedné chyby nebo složitější polynomické kódy jako je CRC (cyclic redundancy code) nebo ECC (error correcting code).

¹ Alexander Ostrowski dokázal v roce 1954, že neexistuje algoritmus na vyhodnocování polynomů, který by používal méně než n sčítání.

BCD kód

Reprezentuje binárně kódovanou decimální číslici. Každá desítková číslice je kódovaná čtyřmi bity v soustavě binární. Číslo zapsané v BCD kódu musí mít počet bitů dělitelný dvěma. U jednotlivých bitů jsou váhy 8-4-2-1.

$25809_D = 0010\ 0101\ 1000\ 0000\ 1001_{BCD}$

Aikenův kód

Stejný jako BCD kód (reprezentuje binárně kódovanou decimální číslici kódovanou na čtyři bity; číslo zapsané v Aik musí mít počet bitů dělitelný čtyřmi) s tím rozdílem, že číslice 0–4 v prvním (nejvyšším) bitě mají 0 a číslice 5–9 v něm mají 1. Hodnoty u jednotlivých bitů jsou pak 2-4-2-1.

$25809_D = 0010\ 1011\ 1110\ 0000\ 1111_{Aik}$

Kód k z n

Umožňuje detekci vzniku jedné chyby při přenosu informace. n je počet bitů (délka slova), k je pak počet jedniček (1).

Například u kódu 2 z 5, byla-li by ověřována informace 10110, by toto znamenalo, že vznikla jedna chyba (počet jedniček je o jedna větší než 2), u kódu 3 z 5 by 10110 vyhovovalo.

ASCII kód

American Standard Code for Information Interchange. Používá se pro zobrazení znaků v počítači. Jeden znak je zobrazen 8 bity (tedy v jednom bajtu). Původně byl sedmibitový, ale byl rozšířen.

Původní část tabulky obsahuje v nejvyšším bitu 0:

Znaky 0–31 jsou řídicí (netisknutelné) znaky.

Znaky 32–127 jsou běžná interpunkční a matematická znaménka, číslice a velká a malá anglická abeceda.

Rozšířená část tabulky obsahuje v nejvyšším bitu 1:

Znaky 128–255 jsou speciální grafické symboly a znaky národních abeced. Znaky národních abeced se mění programově. Např. čeština může být kódována několika způsoby: *bratří Kamenických*, *PC Latin 2*, *Windows-1250 (CP-1250)*, či např. modernější *UTF-8*.

Unicode

Hexadecimální kód je mezinárodní standard. Unicode se používá pro zobrazení znaků v počítači. Prvních 128 znaků Unicode je shodných s ASCII, prvních 256 znaků pak s ISO 8859-1 (které obsahuje také ASCII). Unicode používají nejmodernější textové editory. Původně byl navržen jako 16bitová znaková sada, později se to však ukázalo jako nedostatečné (zejména s ohledem na čínskou znakovou sadu).

Jeden znak je zobrazen 32 bity (tzn. ve čtyřech bajtech). Textové soubory mají větší velikost, ale v kódu jsou obsaženy všechny abecedy (dnes již navíc není větší soubor nijakým problémem). Může zobrazit 2^{32} znaků (tedy 4 294 967 296), v současnosti (verze Unicode 10.0 z roku 2017) je jich však pouze asi 136 690. Všechny tyto znaky jsou z celkem 139 jazyků a abeced.