

## Obsah

<b>2 Informace v počítači</b>	<b>1</b>
2.1 Data v počítači . . . . .	1
2.2 Číselné soustavy . . . . .	1
2.2.1 Dvojková (binární) soustava . . . . .	1
2.2.2 Šestnáctková (hexadecimální) soustava . . . . .	2
2.2.3 Osmičková (oktalová) soustava . . . . .	3
2.3 Jednotky informace . . . . .	4
2.4 Záznamová média . . . . .	4

## 2 Informace v počítači

### 2.1 Data v počítači

- reprezentace pomocí 1 a 0 (zapnuto/vypnuto) – bity [b] (kb, Mb, Gb,...)
- seskupovány po 8 – byty [B] (kB, MB, GB,...)
- data uchovány podle různých řádů a systémů – formáty souborů, kódování jazyků (ASCII vs UTF8) atd.
- metadata
  - data o dalších datech / o souboru
  - specifikace MIME typu, čas, permissions, creator, ...
- konverze dat – převod mezi formáty
  - komprese dat – konverze s cílem zmenšit velikost souborů, lossy/lossless compression
- ukládání – RAM, disky (HDD, SSD, USB disky, paměťové karty), disk array (RAID), cloud

### 2.2 Číselné soustavy

- soustavy specifikující reprezentaci čísel
- báze  $r$  – číslo specifikující počet znaků soustavy (desítková soustava –  $r = 10$ )
- řád  $n$  – hodnota řádů je  $r^n$  (stovky –  $10^2$ )
- hodnota čísla

$$N = \sum_{i=-l}^{k-1} n_i \cdot r^i = n_{k-1}r^{k-1} + n_{k-2}r^{k-2} + \dots + n_0r^0 + n_{-1}r^{-1} + \dots + n_{-l+1}r^{-l+1} + n_{-l}r^{-l}$$

#### 2.2.1 Dvojková (binární) soustava

- báze 2 – čísla 0, 1
  - každá cifra odpovídá mocnině dvou

### Převody

#### Dvojková $\rightarrow$ Desítková

- každou cifru vynásobíme danou mocninou základu a hodnoty sečteme

$$(11010)_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = (26)_{10}$$

**Desítková → Dvojková**

- číslo vydělíme 2, zapíšeme zbytek po dělení a dělíme dál, dokud nedělíme nulu

pořadí dělení	7.	6.	5.	4.	3.	2.	1.	číslo
výsledky dělení	1	3	6	13	26	53	107	214
rozklad na součin 2	$2 \cdot 0 + 1$	$2 \cdot 1 + 1$	$2 \cdot 3$	$2 \cdot 6 + 1$	$2 \cdot 13$	$2 \cdot 26 + 1$	$2 \cdot 53 + 1$	$2 \cdot 107$
zbytky dělení	1	1	0	1	0	1	1	0

**Záporná čísla****Přímý kód**

- první bit je znaménkový
  - $(00000001)_2 = (1)_{10}$
  - $(10000001)_2 = (-1)_{10}$
- potřebné odlišné algoritmy pro sčítání a odčítání
- dvě reprezentace nuly

**Inverzní kód**

- záporné číslo zaznamenáno jako binární negace
- dvě nuly
- různé algoritmy pro sčítání a odčítání

**Doplňkový kód**

- záporné číslo zaznamenáno jako binární negace (výměna 0 a 1) + 1
- podle úvodního bitu lze rozeznat kladná/záporná čísla
- využití přetečení
  - $00000000 - 00000001 = 11111111$
- jediná reprezentace nuly
- stejné algoritmy pro sčítání a odčítání
- zachována komutativnost
- používané v počítačích
- příklad: pokud  $00001101$  je binární vyjádření čísla 13, pak 13 se vypočte jako  $\text{NOT}(00001101) + 1 = 11110010 + 1 = 11110011$

$$20_{10} - 13_{10} = 20_{10} + (-13)_{10} = 00010100_2 + 11110011_2 = \\ = 1\ 00000111_2 = 7_{10} \text{ (po odříznutí přeteklého devátého bitu)}$$

**Desetinná čísla**

- tvořeny pomocí zlomků
- mocniny za desetinnou čárkou mají zápornou hodnotu → součet zlomků

$$(0,1011)_2 = 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = \\ = 0 \cdot 1 + 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} + 1 \cdot \frac{1}{16} = \\ = 0,5 + 0,125 + 0,0625 = 0,6875$$

**2.2.2 Šestnáctková (hexadecimální) soustava**

- báze 16 – čísla 1, 2, 3, 4, 5, 6, 7, 8, 9, A(10), B(11), C(12), D(13), E(14), F(15)
- jednoduchý převod mezi binární a hexadecimální soustavou
- častý zápis v programování – 0xE1

## Převody

### Šestnáctková → Desítková

- stejné jako u dvojkové soustavy, pouze násobíme mocninou 16

### Desítková → Šestnáctková

- stejné jako u dvojkové soustavy, pouze dělíme 16

### Šestnáctková → Dvojková

- každé 4 bity (nibble = 4 bity = polovina bytu) odpovídají jednomu řádu hex-soustavy ( $2^4 = 16$ )
  - 1 = 0001
  - 2 = 0010
  - 3 = 0011
  - ...
  - E = 14 = 1110
  - F = 15 = 1111
- z každé jedné cifry určíme dané 4 bity
- 2 cifry hex odpovídají bytu
- příklad: 0xE1 = 1110 0001

### Dvojková → Šestnáctková

- opačný způsob – z každých 4 bitů určíme odpovídající cifru hex
- příklad: 1011 0101 = 0xB5

## 2.2.3 Osmičková (oktalová) soustava

- číselná soustava o základu 8 – čísla 1, 2, 3, 4, 5, 6, 7
- mocnina 2 – jednoduchý převod do binární soustavy
- použití v informatice – Linux permissions

## Převody

### Osmičková → Desítková

- stejné jako u dvojkové soustavy, pouze násobíme mocninou 8

### Desítková → Osmičková

- stejné jako u dvojkové soustavy, pouze dělíme 8

### Osmičková → Dvojková

- každé 3 bity ( $2^3 = 8$ ) odpovídají jedné číslici osmičkové soustavy
  - 1 = 001
  - 2 = 010
  - 3 = 011
  - ...
  - 6 = 110
  - 7 = 111
- z každé cifry určíme 3 cifry binárního čísla a zapíšeme za sebe

**Dvojková → Osmičková**

- opačný způsob – převod každých 3 bitů na odpovídající cifru osmičkové soustavy
- příklad:  $(101001)_2 = (51)_8$

**Osmičková ↔ Šestnáctková**

- 2 kroky – převod do binární (případně desítkové) soustavy a následné zakódování do dané soustavy
- převod přes desítkovou soustavu univerzální pro všechny soustavy

**2.3 Jednotky informace**

- základní jednotka – bit – 1 b
  - uložení jednoho on-off/true-false/1-0 stavu
- shromažďovány do skupiny po 8 – byte – 1 B = 8 b
- násobky
  - desítková soustava
    - \* kilobyty – 1 kB =  $10^3$  B = 1 000 B
    - \* megabyty – 1 MB =  $10^6$  B
    - \* gigabyty – 1 GB =  $10^9$  B
    - \* terabyty – 1 TB =  $10^{12}$  B
  - dvojková soustava
    - \* kibibyte – 1 KiB =  $2^{10}$  B = 1 024 B
    - \* mebibyte – 1 MiB =  $2^{10}$  KiB =  $2^{20}$  B
    - \* gibibyte – 1 GiB =  $2^{10}$  MiB =  $2^{30}$  B
    - \* tebibyte – 1 TiB =  $2^{10}$  GiB =  $2^{40}$  B

**2.4 Záznamová média**

- historicky
  - děrné pásy – záznam dat pomocí děr
  - diskety (floppy disk) – magnetické úložiště
- CD/DVD – optické disky / malá zrcadla, čtení/zápis pomocí disku
- HDD – Hard drive disk (pevný disk)
  - elektromagnetické úložiště
  - uvnitř točící se disky se čtecí hlavou
  - SATA protokol
  - na dnešní poměry pomalé
  - nízká cena za velkou kapacitu
  - využití – archivace, uchování velkých dat (např. NAS)
- SSD – Solid state drive
  - uchování dat pomocí integrovaného obvodu (flash memory)
  - chybí mechanické části
  - SATA protokol
  - rychlý přístup k datům
  - vyšší cena na jednotku objemu
  - využití – instalace OS, programů, data potřebná rychlé čtení/zápis
- M.2 SSD
  - podobné SATA SSD
  - využívá M.2 slot
  - možné využití NVMe (na PCIe) protokolu – vyšší přenosová rychlost
- USB flash drive
  - flash memory přes USB protokol
  - přenositelná zařízení
  - fyzický přenos dat

- Magnetické pásky
  - uchování dat pomocí magnetismu
  - nízká cena
  - dnešní využití pro archivaci, zálohy