

Otázka číslo 2

Zobrazování informací v počítači

Základní jednotka – bit, byte

- 1 bit (b) = nejmenší jednotka informace
- Dobře se realizuje
 - Pomocí elektrického signálu (není napětí = 0, je napětí = 1)
 - Pomocí mechanických prostředků (není prohlubeň = 0, je prohlubeň = 1)
- Protože ovšem bit není dost velký na zakódování malých a velkých písmen, číslic (a rezervy), vzniknul 1 byte (B) = 8 bitů
 - 8 bitů, kdy každý může nabývat hodnoty 0 či 1, nám vyjde 2^8 možností (=256)
- Skutečnost je zaznamenávána ve formě znaků, které vytváří data. Informace se uchovávají v kódech

Historie

- Baconova šifra z roku 1626

a	AAAAA	g	AABBA	n	ABBAA	t	BAABA
b	AAAAB	h	AABBB	o	ABBAB	u-v	BAABB
c	AAABA	i-j	ABAAA	p	ABBBA	w	BABAA
d	AAABB	k	ABAAB	q	ABBBB	x	BABAB
e	AABAA	l	ABABA	r	BAAAA	y	BABBA
f	AABAB	m	ABABB	s	BAAAB	z	BABBB

- Kódování dat bylo poprvé použito u děrných štítků (médium z kartonu pro záznam dat pro pozdější zpracování automaty nebo počítači) – tuto metodu pak později převzali mj. i společnosti jako IBM
- Tento způsob kódování je například používán u Morseovky
- Jeden z důvodů použití dvojkové soustavy místo například desítkové, protože s ní je práce mnohem komplikovanější

<ul style="list-style-type: none">● Base 10 Number Representation<ul style="list-style-type: none">– Natural representation for human transactions $3.5 = 3 \times 10^0 + 5 \times 10^{-1}$● Hard to Implement Electronically<ul style="list-style-type: none">– Hard to store<ul style="list-style-type: none">• ENIAC (First electronic computer) used 10 vacuum tubes / digit– Hard to transmit<ul style="list-style-type: none">• Need high precision to encode 10 signal levels on single wire– Messy to implement digital logic functions<ul style="list-style-type: none">• Addition, multiplication, etc.● Binary Representation<ul style="list-style-type: none">– Specifically, the devices that make up a computer are switches that can be on or off, i.e. at high or low voltage.– Easy to store with bistable elements– Reliably transmitted on noisy and inaccurate wires
--

Kódování

- Účelem kódování na počítačích je převedení textu na počítačem zpracovatelná data, tedy jedničky a nuly

Způsoby kódování

ASCII

- Původně 7mi bitové (128 znaků), rozšířené na 8 bitů (1 byte), tedy 256 znaků

- Příklad: A – (0)1000001

Unicode

- Několik „podtypů“
 - UTF-8: používá jeden byte pro libovolná (ASCII) znak, používá se pro přenos dat nebo v Linuxu
 - UTF-16: každý znak reprezentován 16bitovým číslem, používán ve Windows pro vnitřní zápis znaků (např jména souborů a adresářů v NTFS (souborový systém))
 - UTF-32: každý znak reprezentován 32bitovým číslem

Násobné jednotky

Pro velikost paměti se často v informatice používají předpony jednotek SI ve významu binární předpony, tedy jako mocnina čísla 2. Tato dvojnásobnost vede ke zmatkům, které bývají mimo jiné účelově využívány v marketingu při prodeji výpočetní techniky.

Ke zmatkům dochází například u kapacity pevných disků, která se běžně uvádí v gigabytech. Mnozí výrobci mají u předpony giga na mysli opravdovou miliardu, i když se v informatice předponou giga často myslí 2^{30} , tedy 1 073 741 824.

Proto Mezinárodní elektrotechnická komise (IEC) doporučuje pro mocniny čísla 2 blízké hodnotám předpon SI používat nové předpony.

Poskytovatelé připojení prakticky výhradně také vyjadřují rychlosti v SI jednotkách (obvykle se uvádí bity místo bajtů). Výrobci pevných disků využívají pro označování SI jednotky, zatímco např. výrobci operačních pamětí používají IEC jednotky (lze to pochopit jako zjednodušení pro spotřebitele, nebo jako jejich klamání), které označují starými SI jednotkami, podobně jako se označují velikosti v prostředí Windows (tedy píše se MB, ale má se na mysli faktický MiB). Je ovšem nutné podotknout, že kromě normy SI a IEC existuje ještě norma JEDEC (sdružení výrobců HW), která se vztahuje zejména na operační paměti a podle které kilobajt představuje 1024 bajtů, megabajt 1024^2 bajtů atd.

Značka	Název	Hodnota
Ki	<i>kibi</i>	1 024
Mi	<i>mebi</i>	1 048 576
Gi	<i>gibi</i>	1 073 741 824
Ti	<i>tebi</i>	1 099 511 627 776
Pi	<i>pebi</i>	1 125 899 906 842 624
Ei	<i>exbi</i>	1 152 921 504 606 846 976
Zi	<i>zibi</i>	1 180 591 620 717 411 303 424
Yi	<i>yobi</i>	1 208 925 819 614 629 174 706 176

Takže:

- SI: jeden kilobajt (kB) = 1000 bajtů
- BP: jeden kibibajt (KiB) = 1024 bajtů
- JEDEC: kilo (K): násobek rovný 1024 → jeden kilobajt = 1024 bajtů

Význam číselné soustavy

- Způsob prezentace čísel
- Ukládání informací – vše se ukládá v dvojkové soustavě jako 1 a 0

Typy soustav

Desítková (decimální, dekadická)

- Čísla se zapisují pomocí 10 číslic (0...9)
- Použití: věda, technika

Příklad

$$1623 = 1 \times 1\,000 + 6 \times 100 + 2 \times 10 + 3 \times 1 = 1 \times 10^3 + 6 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

Dvojková (binární)

- Číslo vyjádřeno pomocí čísla 2 násobených opět jejich pozičními hodnotami, ty však mohou být jen 0 či 1
- Použití: logické členy, ukládání informací

Příklad

$$1623 = 11001010111$$

Šestnáctková (hexadecimální)

- K vyjádření slouží 16 znaků, číslice 0...9 a písmena A(10)...F(15)
- Čísla se obvykle označují písmenem H připojeným k číslu v dolním indexu
- Použití: adresy v operační paměti (RAM) počítače, zápis barev

Příklad

$$1623 = 657 = 6 \times 16^2 + 5 \times 16^1 + 7 \times 16^0$$

$$A0_{16} = 1 \times 16^2 + 0 \times 16^1 + 0 \times 16^0$$

$$A0_{16} = 10 \times 16^1 + 0 \times 16^0$$

$$3A8 = 3 \cdot 16 + 10 \cdot 16 + 8 \cdot 16 = 3 \cdot 256 + 10 \cdot 16 + 8 \cdot 1 = (936)_{10}$$

Barvy, zápis barev

- Zápis slovně či RGB
- Slovně: color="nazev_barvy", barvy se píše pomocí jejich anglických názvů
- Typy RGB zápisu (modrá barva):
 - 1) Procentuální: rgb(0%, 0%, 100%)
 - 2) Desítková: rgb(0,0,255)
 - 3) Šestnáctkový
- Šestnáctkový zápis
 - o Před tento zápis se píše znak #
 - o Za ním následuje 6 znaků
 - o První 2 patří červené, prostřední 2 zelené a poslední 2 modré
 - o Nejvyšší hodnota: FF
 - o Nejnižší hodnota: 00
 - o Modrá barva: #0000FF
 - o Černá barva: #000000
 - o Bílá barva: #FFFFFF

Převod mezi soustavami

Z desítkové do dvojkové

Číslo z desítkové soustavy do libovolné soustavy převedeme tak, že postupně dělíme desítkové číslo základem nové čísl. soustavy a sepisujeme zbytky. Poslední zbytek je nejvyšší řád v nové čísl. Soustavě

```
53D | :2
26 | 1 /\
13 | 0 |
```

```
6 | 1 |
3 | 0 |
1 | 1 |
0 | 1 |
```

53D == 110101B

Z dvojkové do desítkové

Při převodu z binární do desítkové soustavy násobíme jednotlivá čísla odzadu číslem vždy 2x větším než je předchozí počínaje jedničkou tzn. 1,2,4,8,16,32...

```
1*32 + 1*16 + 0*8 + 1*4 + 0*2 + 1*1
==
32 + 16 + 4 + 1
==
53D
```

Z desítkové do šestnáctkové

Převod probíhá úplně stejně jako z desítkové na dvojkovou, akorát prvně číslo dělíme šestnácti a zbytky mohou nabývat až hodnoty 15 (F)

```
1623D | :16
101 | 7 /\
6 | 5 |
0 | 6 |
1623D = 657H
```

Ze šestnáctkové do desítkové

Převod probíhá úplně stejně jako z dvojkové na desítkovou, jen dané číslo v šestnáctkové soustavě násobíme mocninou 16ti na dané pozici

```
6*162 + 5*161 + 7*160
==
1536 + 80 + 7
==
1623D
```

Z dvojkové do šestnáctkové

Číslo si zprava rozdělujeme do skupin po 4 číslicích, pro každou skupinu po čtyřech přiřadíme číslo v desítkové soustavě a tomuto číslu přiřadíme i odpovídající číslo v 16 šestnáctkové soustavě

	10	1110	0001	0100	1101 _B	= 2E14D _H
desítkově	2	14	1	4	13	
šestnáctkově	2	E	1	4	D	

Z šestnáctkové na dvojkovou

Děje se to opačně než z dvojkové na šestnáctkovou

Hex	A	B	3	7
Dec	10	11	3	7
Bin	1010	1011	0011	0111

$$\mathbf{AB37_H = 1010\ 1011\ 0011\ 0111_B}$$

Početni operace

Sčítání v dvojkové soustavě

```

10101100
+11010111
-----
110000011

```

- Sčítáme zprava
- $0+0 = 0$ a $1+0=1$
- V případě $1+1$ píšeme 0 a jedničku započítáme do dalšího sloupce
- Pokud započítáváme 1 do sloupce, kde je $1+1$, zapíšeme 1 a 1 započítáme do dalšího sloupce
- Z toho důvodu může být výsledek o 1 delší

Odčítání v dvojkové soustavě

Způsob 1 – klasické odčítání

- Pokud máme 0 a 0, píšeme 0
- Pokud máme 1 a 1, píšeme 0
- Pokud máme dole 0 a nahoře 1, píšeme 1
- Pokud máme dole 1 a nahoře 0, píšeme 1 a přeneseme 1 do dalšího sloupce do dolní řady (0 se mění na 1 a 1 na 0 s tím, že zase přenáší 1 do další řady)
- Více zde: [https://www.youtube.com/watch?v=SzWSngW1- 4](https://www.youtube.com/watch?v=SzWSngW1-4)

Způsob 2 – přičítání +1 (raději ne)

- Číslo, od kterého odečítáme, zůstane, ale menšiteli napíšeme inverzní číslo ($=0$ se změní na 1 a 1 na 0) a přičteme 1
- Tyto 2 čísla pak sečteme
- Pokud vyjde výsledek delší jak 8 číslic, 1 zleva smažeme

Odčítání ve dvojkové soustavě: **11010110**

$$\begin{array}{r}
 \begin{array}{l}
 - \underline{11100110} \text{ (číslo převedeme na dvojkový doplněk a přičteme} \\
 \text{k vrchnímu). } 11100110 \\
 \Downarrow \\
 00011001 \\
 + \quad 1 \\
 \hline
 00011010
 \end{array} \\
 \\
 \begin{array}{r}
 11010110 \\
 + \underline{00011010} \\
 \hline
 11110000
 \end{array}
 \end{array}$$

Způsob 3

- Číslo, od kterého odečítáme, zůstane, ale menšiteli napíšeme inverzní číslo ($=0$ se změní na 1 a 1 na 0)
- Tyto 2 čísla pak sečteme
- U výsledky pak 1 nejmíc zleva odebereme a přičteme zprava

$$\begin{array}{r}
 0111 \\
 -0010 \\
 \hline
 0111 \\
 +1101 \text{ - Inverzní číslo} \\
 \hline
 10100 \\
 \begin{array}{l} \longleftarrow \\ \longrightarrow \end{array} +1 \\
 \hline
 0101
 \end{array}$$

Sčítání v šestnáctkové soustavě

Příklad: $\frac{5A}{+2F}$

- $A+F = 10+15 = 25 = 15 \rightarrow 0 \rightarrow 1$ zapíšu do dalšího sloupce, zde už jsme použili 15 číslic a 0, tedy 16 z 25, takže nám zbývá 9 a tu zapíšeme
- $5+2+1$ (ta z předchozího sloupce) = 8
- Výsledek je 89
- $\frac{5A}{2F} = 80$

Odčítání v šestnáctkové soustavě

Příklad A: $\frac{5F}{-2A}$

- $F-A = 15-10=5$, zapíšu 5
- $5-2 = 3$, píšu 3
- Výsledek: 35

Příklad B: $\frac{5A}{-2F}$

- A je menší jak F, takže si půjčím 1 od 5 ($5 \rightarrow 4$) a k A tak přičtu 16
- Ted: $(A+16)-F = (10+16)-15 = 26-15 = 11 = B$, píšu B
- $4-2 = 2$, píšu 2
- Výsledek: 2B