



# POČÍTAČE A DATA

DATA, INFORMACE, BIT, BYTE  
ČÍSELNÉ SOUSTAVY

Adam Klepáč

18. září 2023

# OBSAH

Data vs. Informace

Bit a byte

Číselné soustavy

# DATA VS. INFORMACE



## DATUM

**Datum** můžeme vnímat jako reprezentaci faktů, konceptů nebo instrukcí způsobem dostatečně formálním na to, aby mohlo být interpretováno nebo vykonáno člověkem či přístrojem.

# DATA – PŘÍKLADY

## Co můžeme nazvat datem

- „Zítra bude pršet.“ je datum – fakt, který člověk umí interpretovat.

# DATA – PŘÍKLADY

## Co můžeme nazvat datem

- „Zítra bude pršet.“ je datum – fakt, který člověk umí interpretovat.
- „Sečti 2 a 5.“ je datum – instrukce, kterou umí člověk, a v mnoha podobách i počítač, interpretovat.

# DATA – PŘÍKLADY

## Co můžeme nazvat datem

- „Zítra bude pršet.“ je datum – fakt, který člověk umí interpretovat.
- „Sečti 2 a 5.“ je datum – instrukce, kterou umí člověk, a v mnoha podobách i počítač, interpretovat.
- „(255,0,0)“ je většinou datum – koncept, který umí počítač (prostřednictvím překladače programovacího jazyka) interpretovat jako červenou barvu.

# DATA – PŘÍKLADY

## Co datum spíš není

- „1,kuře,juxtapozice,feromon“ – náhodná sada slov a symbolů bez kontextu nic nereprezentuje, přestože každý z nich má odděleně význam.



# DATA – PŘÍKLADY

## Co datum spíš není

- „1,kuře,juxtapozice,feromon“ – náhodná sada slov a symbolů bez kontextu nic nereprezentuje, přestože každý z nich má odděleně význam.
- zemětřesení (obecně jakákoli událost) – údaj o tom, že se událost stala, datem **je**, ta samotná událost **není**.

# DATA – PŘÍKLADY

## Co datum spíš není

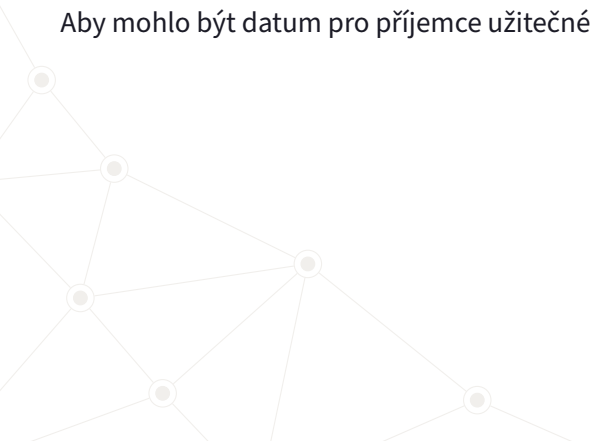
- „1,kuře,juxtapozice,feromon“ – náhodná sada slov a symbolů bez kontextu nic nereprezentuje, přestože každý z nich má odděleně význam.
- zemětřesení (obecně jakákoli událost) – údaj o tom, že se událost stala, datem **je**, ta samotná událost **není**.
- obecně předměty – zase, to, že se s předmětem něco děje, datum **je**, ten samotný předmět **ne**.

## INFORMACE

Informace jsou **data**, která jsou *seřazena* nebo *roztříděna* a mají hodnotu pro příjemce. Jsou to zpracovaná data, jimiž se řídí budoucí činy a rozhodnutí.

# INFORMACE – CHARAKTERISTIKY

Aby mohlo být datum pro příjemce užitečné (a tedy být **informací**), musí být



# INFORMACE – CHARAKTERISTIKY

Aby mohlo být datum pro příjemce užitečné (a tedy být **informací**), musí být

- **včasné** – poskytnuto v době, kdy je stále relevantní;

# INFORMACE – CHARAKTERISTIKY

Aby mohlo být datum pro příjemce užitečné (a tedy být **informací**), musí být

- **včasné** – poskytnuto v době, kdy je stále relevantní;
- **přesné** – neobsahující chyby a víceznačné formulace;

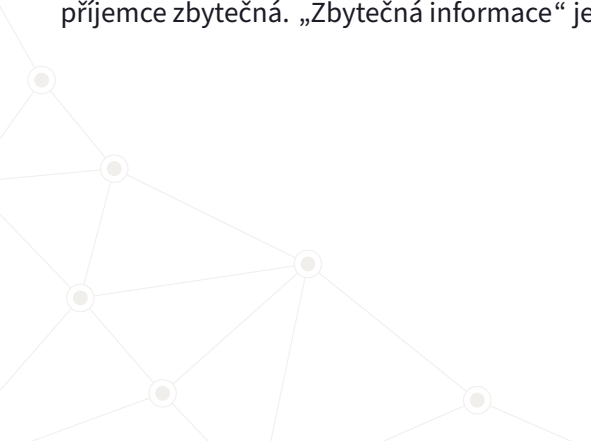
# INFORMACE – CHARAKTERISTIKY

Aby mohlo být datum pro příjemce užitečné (a tedy být **informací**), musí být

- **včasné** – poskytnuto v době, kdy je stále relevantní;
- **přesné** – neobsahující chyby a víceznačné formulace;
- **úplné** – žádná část důležitá pro interpretaci nesmí chybět.

# INFORMACE – PŘÍKLADY

V jazyce občas říkáme „zbytečná informace“. Bacha na to! Informace **z definice** není pro příjemce zbytečná. „Zbytečná informace“ je tudíž prostě **datum**.





# INFORMACE – PŘÍKLADY

V jazyce občas říkáme „zbytečná informace“. Bacha na to! Informace **z definice** není pro příjemce zbytečná. „Zbytečná informace“ je tudíž prostě **datum**.

## Kdy datum je a kdy není informací?

- V moment, kdy nakupuji, je datum ceny produktu informací.

# INFORMACE – PŘÍKLADY

V jazyce občas říkáme „zbytečná informace“. Bacha na to! Informace **z definice** není pro příjemce zbytečná. „Zbytečná informace“ je tudíž prostě **datum**.

## Kdy datum je a kdy není informací?

- V moment, kdy nakupuji, je datum ceny produktu informací.
  - Je **včasné**? ✓ Nakupuji *ted'*.

# INFORMACE – PŘÍKLADY

V jazyce občas říkáme „zbytečná informace“. Bacha na to! Informace **z definice** není pro příjemce zbytečná. „Zbytečná informace“ je tudíž prostě **datum**.

## Kdy datum je a kdy není informací?

- V moment, kdy nakupuji, je datum ceny produktu informací.
  - Je **včasné**? ✓ Nakupuji *ted'*.
  - Je **přesné**? ✓ Nemůže se stát, že bych produkt koupil za jinou cenu.

# INFORMACE – PŘÍKLADY

V jazyce občas říkáme „zbytečná informace“. Bacha na to! Informace **z definice** není pro příjemce zbytečná. „Zbytečná informace“ je tudíž prostě **datum**.

## Kdy datum je a kdy není informací?

- V moment, kdy nakupuji, je datum ceny produktu informací.
  - Je **včasné**? ✓ Nakupuji *ted'*.
  - Je **přesné**? ✓ Nemůže se stát, že bych produkt koupil za jinou cenu.
  - Je **úplné**? ✓ Ano, číslo vyjadřuje absolutní hodnotu.

# INFORMACE – PŘÍKLADY

## Kdy datum je a kdy není informací?

- Vzkaz „Supluješ první hodinu v pondělí.“ řečený v úterý odpoledne *není* informací. Je sice **přesný**, ale není ani **včasný** ani **úplný**.

# INFORMACE – PŘÍKLADY

## Kdy datum je a kdy není informací?

- Vzkaz „Supluješ první hodinu v pondělí.“ řečený v úterý odpoledne *není* informací. Je sice **přesný**, ale není ani **včasný** ani **úplný**.
- Příkaz „Sečti nebo vyděl dvě čísla.“ není informací. Je sice **včasný**, ale ani **přesný** ani **úplný**.

# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)



V nejširším slova smyslu jsou počítače stroje, které **data** převádějí na **informace**.



# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)

V nejširším slova smyslu jsou počítače stroje, které **data** převádějí na **informace**.  
Každý počítačový program je **cyklus zpracování dat**.



# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)

V nejširším slova smyslu jsou počítače stroje, které **data** převádějí na **informace**.  
Každý počítačový program je **cyklus zpracování dat**.



# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)



# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)



- **vstup** (input) – data ve formě umožňující zpracování

# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)



- **vstup** (input) – data ve formě umožňující zpracování
- **zpracování** (processing) – data jsou transformována v data užitečnější

# DATUM → INFORMACE (DATA PROCESSING/ZPRACOVÁNÍ DAT)



- **vstup** (input) – data ve formě umožňující zpracování
- **zpracování** (processing) – data jsou transformována v data užitečnější
- **výstup** (output) – transformovaná jsou sesbírána a předána jako informace

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

Náš mozek zpracovává data neustále.

- Smysly – paprsky světla nebo vibrace ve vzduchu jsou **data**, nejsou nám nijak užitečná. Smysly jsou způsob, jak náš mozek dostává **vstup**. **Výstupem** je jakýsi model našeho okolí.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Náš mozek zpracovává data neustále.

- Smysly – paprsky světla nebo vibrace ve vzduchu jsou **data**, nejsou nám nijak užitečná. Smysly jsou způsob, jak náš mozek dostává **vstup**. **Výstupem** je jakýsi model našeho okolí.
- Logika – vzkaz „Zastávka na znamení.“ je **datum**, není v užitečné formě. Je **vstupem**, který vyhodnotíme (**zpracujeme**) a vyrobíme z něj **výstup** – informaci, že musíme zmáchnout tlačítko, abychom směli vystoupit.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.



# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.
  - **Zpracování** spočívá v provedení operace součtu.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.
  - **Zpracování** spočívá v provedení operace součtu.
  - **Výstupem** je číslo 5.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.
  - **Zpracování** spočívá v provedení operace součtu.
  - **Výstupem** je číslo 5.
- Otevření aplikace.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.
  - **Zpracování** spočívá v provedení operace součtu.
  - **Výstupem** je číslo 5.
- Otevření aplikace.
  - **Vstupem** jsou dvě kliknutí na ikonku.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.
  - **Zpracování** spočívá v provedení operace součtu.
  - **Výstupem** je číslo 5.
- Otevření aplikace.
  - **Vstupem** jsou dvě kliknutí na ikonku.
  - **Zpracování** spočívá v otevření aplikace s odpovídající ikonkou.

# CYKLUS ZPRACOVÁNÍ DAT – PŘÍKLADY

## Ted' něco s počítačem...

- Sečti čísla 2 a 3.
  - **Vstupem** jsou čísla 2 a 3.
  - **Zpracování** spočívá v provedení operace součtu.
  - **Výstupem** je číslo 5.
- Otevření aplikace.
  - **Vstupem** jsou dvě kliknutí na ikonku.
  - **Zpracování** spočívá v otevření aplikace s odpovídající ikonkou.
  - **Výstupem** je okno s otevřenou aplikací.

BIT A BYTE



# CO JE BIT

## BIT

**Bit** (binary digit) je nejmenší datová jednotka, která může být uložena v paměti počítače.

# JEDNOTKY POČÍTAČOVÉ PAMĚTI

Název	Popis
bit	Logická 0 („ne“) nebo 1 („ano“) představující stav komponenty elektrického obvodu.
nibble	Skupina 4 bitů. <b>Nepoužívá se.</b>
byte (B)	Skupina 8 bitů.
slovo (word)	Skupina fixního počtu bitů. Různá procesor od procesoru (obvykle mezi 8 a 96 bity).

# JEDNOTKY POČÍTAČOVÉ PAMĚTI

Název	Popis
<b>kilobyte (KB)</b>	1024 bytů.
<b>megabyte (MB)</b>	1024 KB = 1048576 B.
<b>gigabyte (GB)</b>	1024 MB = 1073741824 B.
<b>terabyte (TB)</b>	1024 GB = 1099511628000 B.

# ČÍSELNÉ SOUSTAVY

# HODNOTA ČÍSLICE

Počítače umějí interpretovat pouze čísla. Hodnota každé číslice v čísle je dána



# HODNOTA ČÍSLICE

Počítače umějí interpretovat pouze čísla. Hodnota každé číslice v čísle je dána

- tou číslicí,

# HODNOTA ČÍSLICE

Počítače umějí interpretovat pouze čísla. Hodnota každé číslice v čísle je dána

- tou číslicí,
- pozicí této číslice v čísle,

# HODNOTA ČÍSLICE

Počítače umějí interpretovat pouze čísla. Hodnota každé číslice v čísle je dána

- tou číslicí,
- pozicí této číslice v čísle,
- základem číselné soustavy (počtu číslic, které jsou v dané soustavě k dispozici).



# HODNOTA ČÍSLICE – PŘÍKLAD

5 1 2 6 7

číslice: 2,  
pozice: **stovky**,  
soustava: 10

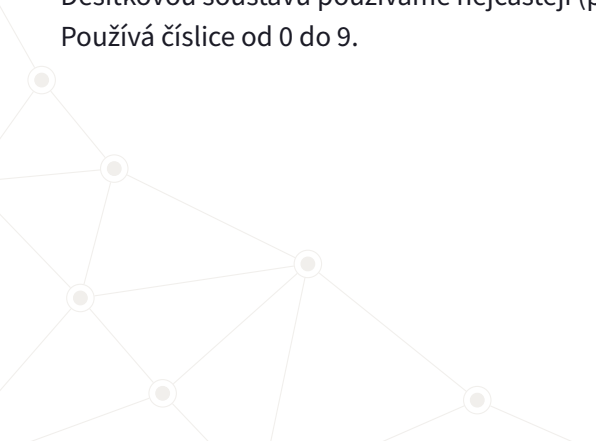
# DESÍTKOVÁ SOUSTAVA

Desítkovou soustavu používáme nejčastěji (protože máme deset prstů).



# DESÍTKOVÁ SOUSTAVA

Desítkovou soustavu používáme nejčastěji (protože máme deset prstů).  
Používá číslice od 0 do 9.



# DESÍTKOVÁ SOUSTAVA

Desítkovou soustavu používáme nejčastěji (protože máme deset prstů).

Používá číslice od 0 do 9.

- Každé číslo v desítkové soustavě lze rozložit na součet násobků mocnin čísla 10.

# DESÍTKOVÁ SOUSTAVA

Desítkovou soustavu používáme nejčastěji (protože máme deset prstů).

Používá číslice od 0 do 9.

Každé číslo v desítkové soustavě lze rozložit na součet násobků mocnin čísla 10. Například:

$$\begin{aligned} 4321 &= (4 \cdot 1000) + (3 \cdot 100) + (2 \cdot 10) + (1 \cdot 1) \\ &= (4 \cdot 10^3) + (3 \cdot 10^2) + (2 \cdot 10^1) + (1 \cdot 10^0). \end{aligned}$$

# DALŠÍ SOUSTAVY BĚŽNÉ V INFORMATICE

Název	Popis
dvojková (binární)	Základ <b>2</b> . Číslice 0 a 1.
osmičková (oktální)	Základ <b>8</b> . Číslice od 0 do 7.
šestnáctková (hexadecimální)	Základ <b>16</b> . Číslice od 0 do 9 a písmena od A do F.

# PŘÍKLAD – SČÍTÁNÍ V OSMIČKOVÉ SOUSTAVĚ

5	1	2	6	7
1	2	1	4	6

# PŘÍKLAD – SČÍTÁNÍ V OSMIČKOVÉ SOUSTAVĚ

$$\begin{array}{r}
 5 \ 1 \ 2 \ 6 \ 7 \\
 + \ 1 \ 2 \ 1 \ 4 \ 6 \\
 \hline
 5
 \end{array}$$

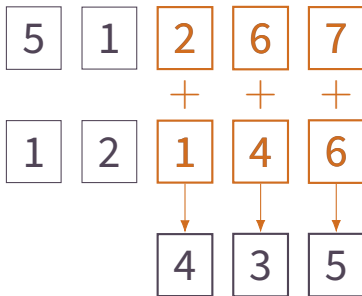
The diagram illustrates an addition in base 8. The first number is 51267 (base 8) and the second number is 12146 (base 8). The sum is 5 (base 8). The digits 7, 6, and 5 are highlighted in orange, and an orange arrow points from the 6 in the second row to the 5 in the result box.



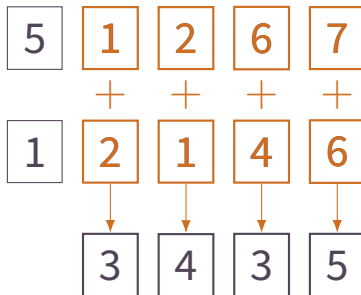
# PŘÍKLAD – SČÍTÁNÍ V OSMIČKOVÉ SOUSTAVĚ

5	1	2	6	7
			+	+
1	2	1	4	6
			↓	↓
			3	5

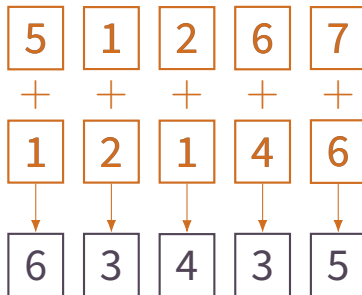
# PŘÍKLAD – SČÍTÁNÍ V OSMIČKOVÉ SOUSTAVĚ



# PŘÍKLAD – SČÍTÁNÍ V OSMIČKOVÉ SOUSTAVĚ



# PŘÍKLAD – SČÍTÁNÍ V OSMIČKOVÉ SOUSTAVĚ



# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ

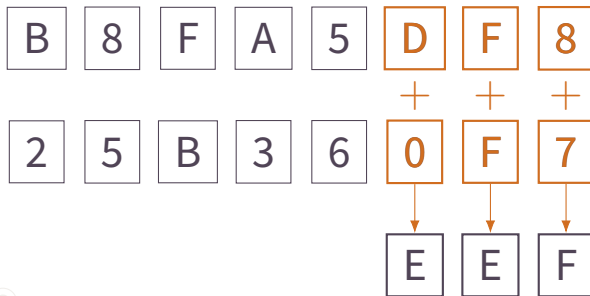
B	8	F	A	5	D	F	8
2	5	B	3	6	0	F	7

# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ

B	8	F	A	5	D	F	8
2	5	B	3	6	0	F	7
							+
							↓
							F



# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ

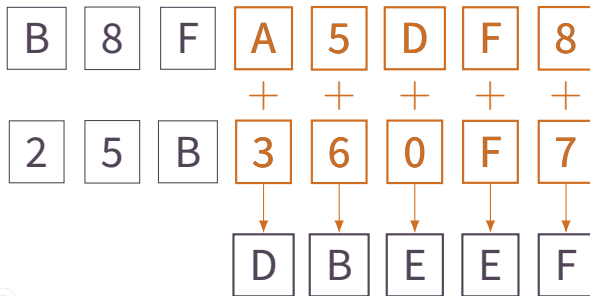




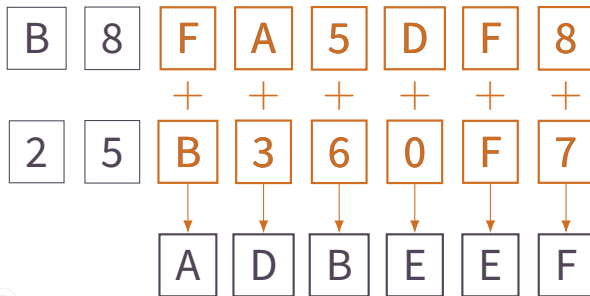
# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ

B	8	F	A	5	D	F	8
				+	+	+	+
2	5	B	3	6	0	F	7
				↓	↓	↓	↓
				B	E	E	F

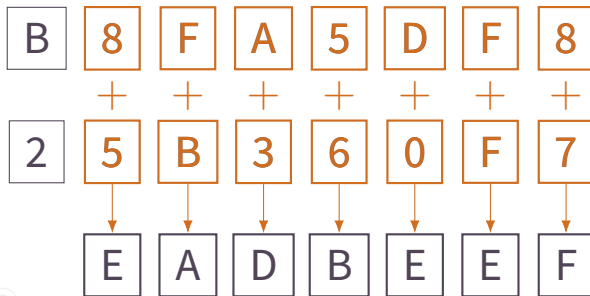
# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ



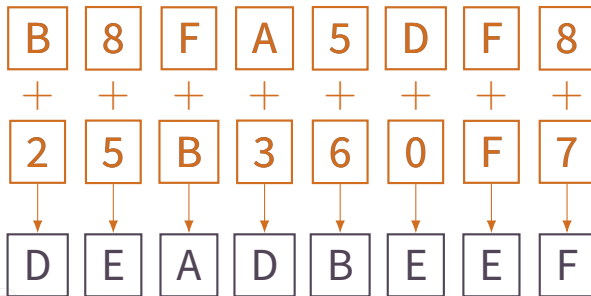
# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ



# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ



# PŘÍKLAD – SČÍTÁNÍ V ŠESTNÁCTKOVÉ SOUSTAVĚ



# PŘEVOD COKOLIV $\rightarrow$ DESÍTKOVÁ

Když je třeba, píšeme u čísel soustavu v dolním indexu.



# PŘEVOD COKOLIV $\rightarrow$ DESÍTKOVÁ

Když je třeba, píšeme u čísel soustavu v dolním indexu.

Například  $4056_{10}$  je číslo 4056 v desítkové soustavě,  $A4578FB_{16}$  je číslo  $A4578FB$  v šestnáctkové soustavě a  $1000101_2$  je číslo 1000101 dvojkové soustavě.

# PŘEVOD COKOLIV $\rightarrow$ DESÍTKOVÁ

Když je třeba, píšeme u čísel soustavu v dolním indexu.

Například  $4056_{10}$  je číslo 4056 v desítkové soustavě,  $A4578FB_{16}$  je číslo  $A4578FB$  v šestnáctkové soustavě a  $1000101_2$  je číslo 1000101 dvojkové soustavě.

- Číslo se v každé soustavě **rozkládá na součet násobků mocnin 10**.
- Abychom převedli číslo z libovolné soustavy do desítkové, stačí tedy číslo 10 převést do desítkové soustavy a pak násobky mocnin sečíst.
- To je snadné, protože 10 je vždy základ soustavy.



# PŘEVOD COKOLIV → DESÍTKOVÁ – PŘÍKLAD

Vezměme třeba číslo  $10011101_2$ . To se v dvojkové soustavě rozkládá jako

$$\begin{aligned} 10011101_2 &= 1 \cdot 10_2^7 + 0 \cdot 10_2^6 + 0 \cdot 10_2^5 + 1 \cdot 10_2^4 + 1 \cdot 10_2^3 + 1 \cdot 10_2^2 + 0 \cdot 10_2^1 + 1 \cdot 10_2^0 \\ &= 10_2^7 + 10_2^4 + 10_2^3 + 10_2^2 + 10_2^0. \end{aligned}$$

# PŘEVOD COKOLIV → DESÍTKOVÁ – PŘÍKLAD

Vezměme třeba číslo  $10011101_2$ . To se v dvojkové soustavě rozkládá jako

$$\begin{aligned} 10011101_2 &= 1 \cdot 10_2^7 + 0 \cdot 10_2^6 + 0 \cdot 10_2^5 + 1 \cdot 10_2^4 + 1 \cdot 10_2^3 + 1 \cdot 10_2^2 + 0 \cdot 10_2^1 + 1 \cdot 10_2^0 \\ &= 10_2^7 + 10_2^4 + 10_2^3 + 10_2^2 + 10_2^0. \end{aligned}$$

Ale,  $10_2 = 2_{10}$ . Čili, v desítkové soustavě je to číslo

$$2^7 + 2^4 + 2^3 + 2^2 + 2^0 = 157.$$

# PŘEVOD COKOLIV → DESÍTKOVÁ – PŘÍKLAD

Vezměme třeba číslo  $10011101_2$ . To se v dvojkové soustavě rozkládá jako

$$\begin{aligned} 10011101_2 &= 1 \cdot 10_2^7 + 0 \cdot 10_2^6 + 0 \cdot 10_2^5 + 1 \cdot 10_2^4 + 1 \cdot 10_2^3 + 1 \cdot 10_2^2 + 0 \cdot 10_2^1 + 1 \cdot 10_2^0 \\ &= 10_2^7 + 10_2^4 + 10_2^3 + 10_2^2 + 10_2^0. \end{aligned}$$

Ale,  $10_2 = 2_{10}$ . Čili, v desítkové soustavě je to číslo

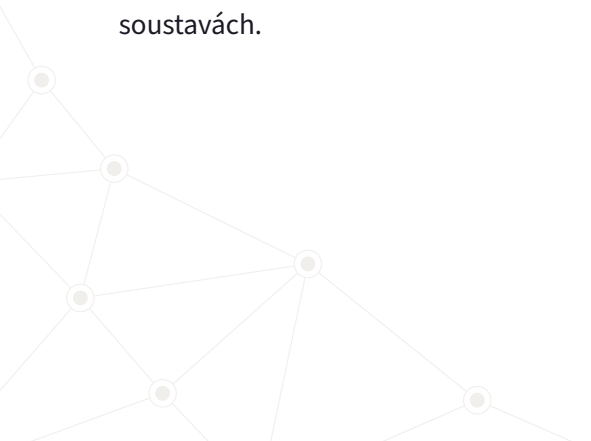
$$2^7 + 2^4 + 2^3 + 2^2 + 2^0 = 157.$$

Takže,

$$10011101_2 = 157_{10}.$$

# PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV

- Opačný postup nebude efektivní, protože neumíme rychle počítat v jiných soustavách.



# PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV

- Opačný postup nebude efektivní, protože neumíme rychle počítat v jiných soustavách.
- Místo toho využijeme tzv. *Hornerova schématu*.

# PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV

- Opačný postup nebude efektivní, protože neumíme rychle počítat v jiných soustavách.
- Místo toho využijeme tzv. *Hornerova schématu*.
- To je způsob, jak rozdělit dané číslo na součet násobků mocnin libovolného jiného čísla „opakovaným vytýkáním“.

# PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV

- Opačný postup nebude efektivní, protože neumíme rychle počítat v jiných soustavách.
- Místo toho využijeme tzv. *Hornerova schématu*.
- To je způsob, jak rozdělit dané číslo na součet násobků mocnin libovolného jiného čísla „opakovaným vytýkáním“.
- Při převodu z desítkové soustavy do soustavy se základem  $a$  tedy stačí pořád dokola z tohoto čísla vytýkat  $a$ , dokud se nedostanu na 1.

## PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV – PŘÍKLAD

Převédeme číslo  $49_{10}$  do dvojkové soustavy, tím že z něj budeme opakovaně vytýkat dvojku.





# PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV – PŘÍKLAD

Převodíme číslo  $49_{10}$  do dvojkové soustavy, tím že z něj budeme opakovaně vytýkat dvojku.

$$\begin{aligned}
 49 &= 2 \cdot (24) + 1 \\
 &= 2 \cdot (2 \cdot (12)) + 1 \\
 &= 2 \cdot (2 \cdot (2 \cdot (6))) + 1 \\
 &= 2 \cdot (2 \cdot (2 \cdot (2 \cdot (3)))) + 1 \\
 &= 2 \cdot (2 \cdot (2 \cdot (2 \cdot (2 \cdot (1) + 1)))) + 1 \\
 &= 2^5 + 2^4 + 2^0 \\
 &= 10_2^5 + 10_2^4 + 10_2^0 \\
 &= 1 \cdot 10_2^5 + 1 \cdot 10_2^4 + 0 \cdot 10_2^3 + 0 \cdot 10_2^2 + 0 \cdot 10_2^1 + 1 \cdot 10_2^0 = 110001_2.
 \end{aligned}$$

# PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV – PŘÍKLAD

Ten samý postup se přehledněji zapíše do tabulky.



## PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV – PŘÍKLAD

Ten samý postup se přehledněji zapíše do tabulky.

Totíž, opakovaně vytýkat dvojku vlastně znamená opakovaně dělit číslo 2 se zbytkem.



## PŘEVOD DESÍTKOVÁ $\rightarrow$ COKOLIV – PŘÍKLAD

Ten samý postup se přehledněji zapíše do tabulky.

Totíž, opakovaně vytýkat dvojku vlastně znamená opakovaně dělit číslo 2 se zbytkem. Pro číslo 49 vypadá tabulka následovně.

Podíl	Zbytek
24	1
12	0
6	0
3	0
1	1
0	1

## PŘEVOD DESÍTKOVÁ → COKOLIV – PŘÍKLAD

Ten samý postup se přehledněji zapíše do tabulky.

Totíž, opakovaně vytýkat dvojku vlastně znamená opakovaně dělit číslo 2 se zbytkem. Pro číslo 49 vypadá tabulka následovně.

Podíl	Zbytek
24	1
12	0
6	0
3	0
1	1
0	1

Když sloupec se zbytky přečteme zezdola nahoru, objevíme číslo 110001, což je číslo 49 ve dvojkové soustavě.