Informace, kódování a redundance

- **Data** (jednotné číslo **údaj**) obvykle chápeme jako údaje, tj. číselné hodnoty, znaky, texty a další fakta zaznamenaná (a uložená v databázi) ve formě uspořádané posloupnosti znaků zvolené abecedy. Jedná se tedy o jakékoli vyjádření (reprezentaci) skutečnosti, schopné přenosu, uchování, interpretace či zpracování.

- Pojem **informace** patří k nejobecnějším kategoriím současné vědy i filozofie. Podle toho, v kterém vědním oboru nebo v které oblasti lidské činnosti se používá, jsou aplikovány specifické přístupy ke zkoumání informace a jsou k dispozici různé způsoby jejího definování
  - V nejobecnějším slova smyslu se informací chápe údaj o reálném prostředí, o jeho stavu a procesech v něm probíhajících. Informace snižuje nebo odstraňuje neurčitost systému (např. příjemce informace); množství informace je dáno rozdílem mezi stavem neurčitosti systému (entropie[1]), kterou měl systém před přijetím informace a stavem neurčitosti, která se přijetím informace odstranila. V tomto smyslu může být informace považována jak za vlastnost organizované hmoty vyjadřující její hloubkovou strukturu (varietu), tak za produkt poznání fixovaný ve znakové podobě v informačních nosičích.
  - V informační vědě a knihovnictví se informací rozumí především sdělení, komunikovatelný poznatek, který má význam pro příjemce nebo údaj usnadňující volbu mezi alternativními rozhodovacími možnostmi. Významné pro informační vědu je také pojetí informace jako psychofyziologického jevu a procesu, tedy jako součásti lidského vědomí (např. N. Wiener[2] definuje informaci jako "obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj svým přizpůsobováním").
  - V exaktní vědě se např. za informaci považuje sdělení, které vyhovuje přísným kriteriím logiky či příslušné vědy.
  - V ekonomické vědě se informací rozumí sdělení, jehož výsledkem může být zisk nebo užitek.

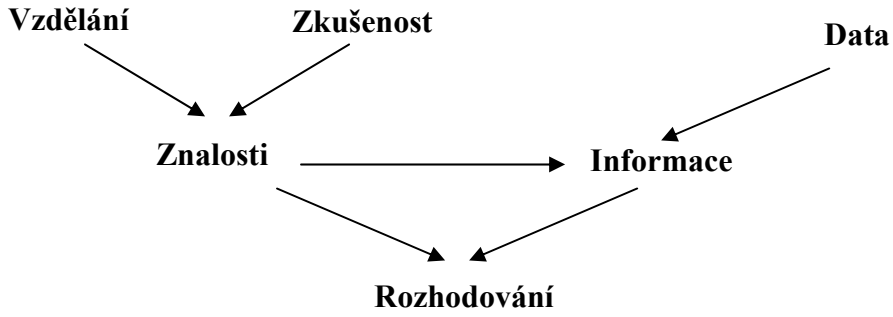
- **Znalost** je to, co jednotlivec vlastní (ví) po osvojení dat a informací a po jejich začlenění do souvislostí. Znalost je tedy výsledek poznávacího procesu a předpoklad uvědomělé činnosti. Všimněme si rozdílu vycházejícího z rozboru rozhodovacího procesu, v němž expert vybavený znalostmi zvažuje data a informace relevantní pro daný problém. Znalosti expert získal vzděláním a zkušenostmi a vybraná data a informace se týkají konkrétní situace či případu.

- **Komplexní poznání** je pak možno považovat za množinu znalostí, informací a dat vztahujících se k určité problematice.

[1] **Entropie** je neurčitost, nejistota, neuspořádanost; střední hodnota míry informace potřebné k odstranění neurčitosti, která je dána konečným počtem vzájemně se vylučujících jevů.

[2] **Norbert Wiener** (26.11.1894 - 18.3.1964) - americký matematik, zakladatel kybernetiky. Zabýval se matematickou analýzou, teorií pravděpodobnosti, matematickou statistikou a výpočetní technikou. Je spoluautorem teorie podobnosti činnosti nervové soustavy a počítače, základu neurokybernetiky.

http://ii.fch.cz



- **Počítač (výpočetní systém)** je stroj na automatické ukládání, zpracovávání, zpřístupňování a přenos dat (automatizuje informační činnosti).

Jednotky informace

- V oblasti výpočetní techniky se za informaci považuje **kvantitativní vyjádření obsahu zprávy**.

- Za jednotku informace se ve výpočetní technice považuje rozhodnutí mezi dvěma alternativami a vyjadřuje se jednotkou nazvanou **bit** (*binary digit*). Informace velikosti jednoho bitu může nabývat hodnot:
  - **ANO** (true)
  - **NE** (false)

- Bitu odpovídají hodnoty **1 = ANO**, **0 = NE** v dvojkové (binární) soustavě.
  - **Téměř** všechny současné počítače pracují v binární soustavě. Technicky je totiž relativně snadné reprezentovat dva rozdílné stavy, např. úrovní elektrického napětí.

- **Bit je dále nedělitelný.**

- **BYTE** (*čti bajt*) = **8 bitů** (256 možností)
  - binárně vyjádřená čísla 0 – 255
  - Pokusy o český ekvivalent (**slabika** ?, **znak** ?) byly neúspěšné
  - **Půlbyte – 4 bity** (16 možností) – číslice v šestnáctkové (hexadecimální) soustavě. Byte je pak vyjádřen dvojicí šestnáctkových číslic
    - {0,1,2,3,4,5,6,7,8,9,A=10,B=11,C=12,D=13,E=14,F=15}

- **Značení: b = bit** *(čti bit)* **B = byte** *(čti bajt)*

- **SLOVO** podle typu počítače většinou **2 nebo 4 byty**

  - Byty v 4-bytovém slově mohou být číslována od 0 do 3 buď **zleva doprava** nebo **zprava doleva**.

  - **Big endian**
    - zleva doprava
    - IBM, SPARK, MOTOROLA

0	1	2	3

http://kip.pef.czu.cz

- **Little endian**
  - zprava doleva
  - INTEL (Pentium)

3	2	1	0

- **Násobky jednotek** (*předpony z latiny*)

Předpona	Značení	Fyzikální jednotky	Informatické jednotky
kilo-	k	$10^3 = 1.000$	$2^{10} = 1.024$
mega-	M	$10^6 = 1.000.000$	$2^{20} = 1.048.576$
giga-	G	10^9	2^{30}
tera-	T	10^{12}	2^{40}
…			

- **Zlomky** (*předpony z řečtiny*)
  - ***mili-*** **,** ***mikro-*****,** ***piko-*****,** …
  - užívané pro fyzikální jednotky, například pro čas a vzdálenost, nemají pro informatické jednotky smysl.

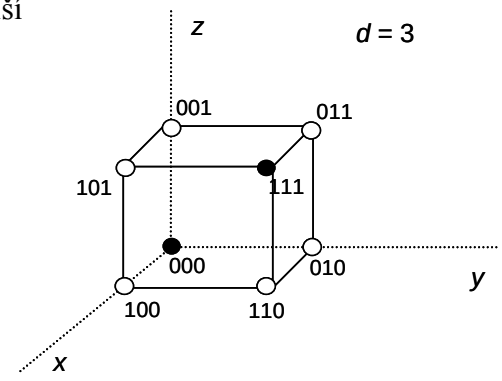
Kódování a redundance dat

- Při ukládání, zpracování a přenosu dat je nutným požadavkem **SPOLEHLIVOST**
- Jeden z prostředků dosažení spolehlivosti je **REDUNDANCE** = **NADBYTEČNOST** dat
- Data jsou v počítači **kódována**
- **Kód** ≡ **prosté (vzájemně jednoznačné) zobrazení množiny kódovaných objektů** $\{O_j\}$ **do množiny kódových slov** $\{K_j\}$ *(v informatice obvykle posloupností 0 a 1)*
  - $\{O_j\} \rightarrow \{K_j = k(O_j)\}, O_i \neq O_j \Rightarrow k(O_i) \neq k(O_j)$

Hammingova vzdálenost

- **Hammingova vzdálenost kódových slov** $X = (x_1, x_2, \ldots, x_n)$ a $Y = (y_1, y_2, \ldots, y_n)$
  - počet míst, na kterých se kódová slova liší

$$d(X,Y) = \sum_{i=1}^{n} |x_i - y_i|$$



http://kip.pef.czu.cz

- **Minimální Hammingova vzdálenost kódu *K***
  - *nejmenší ze vzdáleností všech dvojic různých kódových slov*

$$d_{min} = \min_{X \neq Y \in K} d(X,Y)$$

- **Poznámka:** Hammingova vzdálenost tvoří z množiny kódových slov metrický prostor = množina se vzdáleností d pro kterou platí:
  - $d(x,y) \geq 0$ a $d(x,y) = 0$ pouze když $x = y$
  - $d(x,y) = d(y,x)$ pro každé x a y
  - $d(x,y) \geq d(x,z) + d(z,y)$ pro každé x, y, z

Typy kódů podle úrovně zabezpečení

- **Zabezpečující kód**
  - při změně jednoho bitu (z 0 na 1 nebo z 1 na 0) nemůže dojít k záměně za jiné kódové slovo (chyba v $d_{min} - 1$ bitech se detekuje).

$$d_{min} \geq 2$$

  - pokud dojde k chybě ve více bitech, může k záměně dojít

- **Samoopravný kód** (*single error corected* - **SEC kód)**
  - při změně jednoho bitu (z 0 na 1 nebo z 1 na 0) nemůže dojít k záměně za jiné kódové slovo a lze určit, které slovo bylo pokaženo (chyba ve $(d_{min} - 1)$ **div 2** se oprav

$$d_{min} \geq 3$$

  - pokud dojde k chybě ve dvou a více bitech, může dojít k chybné opravě

- **SEC DED kód** (*single error corrected, double error detected*)
  - chyba v jednom bitu se opraví, chyba ve dvou bitech se detekuje.

- **Kódy s vyšší úrovní zabezpečení** – detekují, případně jsou schopné opravit i chyby ve více bitech dané jednotky dat. **Vyžadují větší redundanci**.

Konstrukce zabezpečujícího kódu

- Zabezpečující kód lze získat přidáním jednoho paritního bitu, tak aby počet jedniček byl
  - buď **sudý** = **sudá parita**
    - $(x_1, \ldots, x_n) \rightarrow (x_1, \ldots, x_n, x_{n+1})$,
    - $\sum_{i=1}^{n+1} x_i = 0, \; x_{n+1} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$
  - nebo **lichý** = **lichá parita**
    - $(x_1, \ldots, x_n) \rightarrow (x_1, \ldots, x_n, x_{n+1})$,
    - $\sum_{i=1}^{n+1} x_i = 1, \; x_{n+1} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \oplus 1$

- Zabezpečující kód získáme za cenu **redundance jediného bitu.**

http://kiwi.fei.czu.cz

Funkce XOR

- Funkce ⊕ = **XOR** = **non ⇔**
  - exclusive OR = "vylučující nebo" = antiekvivalence

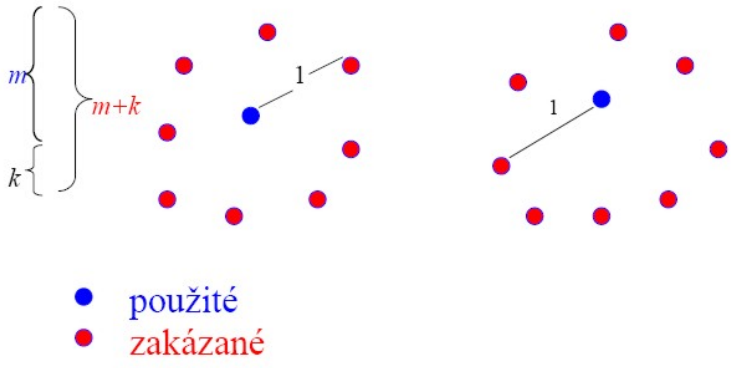
- Dává hodnotu
  - **1 = *TRUE*** když jeden a **jen jeden argument** je 1 = *TRUE*
  - **0 = *FALSE*** když jsou **oba argumenty** 0 = *FALSE* nebo oba argumenty 1 = *TRUE*

- Pravdivostní tabulka funkce:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Redundance pro samoopravný (SEC) kód

- **Odvození:**
  - binární kód **délky *m* bitů**
  - **minimální Hammingova vzdálenost rovna 1**.
    - umožní kódovat **2^m slov**.
  - Doplňme ***k*** bitů s požadavkem, aby kód délky ***m+k*** byl již **zabezpečující**.
    - to dává **2^{m+k} možností**.
  - V "**okolí**" každého z **2^m použitých** je ***m+k*** "**zakázaných**" (ty totiž mají Hammingovu vzdálenost 1).



$$2^m + (m+k) \cdot 2^m \leq 2^{(m+k)}$$

$$k + m + 1 \leq 2^k$$

http://kii.pef.czu.cz

- **Obecně** pro opravu *k* chyb v jednom slově je třeba aby: $d_{min} = 2 \cdot k + 1$
  - Jinými slovy: Automaticky lze vždy opravit současně $[(d_{min} - 1)/2]$ **chyb,** kde […] označuje celou část čísla

Tabulka nutné nadbytečnosti pro SEC kód

Délka kódu v bitech	Třeba přidat minimálně bitů	Redundance kódu v %
2	3	150
4	3	75
8 = 1B	4	50
16 = 2B	5	31
32 = 4B	6	19
64 = 8B	7	11
128 = 16B	8	7
256 = 32B	9	3

- Čím delší úsek dat zabezpečujeme jako celek, tím menší je procentuální nadbytečnost kódu a tedy lepší využití paměti.

- *ALE (něco za něco, aneb krajíc chleba má dvě strany)*
  - Čím delší úsek dat zabezpečujeme jako celek, tím větší je pravděpodobnost, že v tomto úseku dojde k více než jedné chybě.

- Ke získání **SEC-DED** stačí doplnit při $d_{min} = 3$ **jediný paritní bit celkové parity**. Pak se chyba ve dvou bitech detekuje (ale neopraví).

Příklad konstrukce SEC kódu

- Je dán **8-bitový** kód s $d_{min} = 1$, je třeba jej rozšířit na **samoopravný** kód s $d_{min} \geq 3$

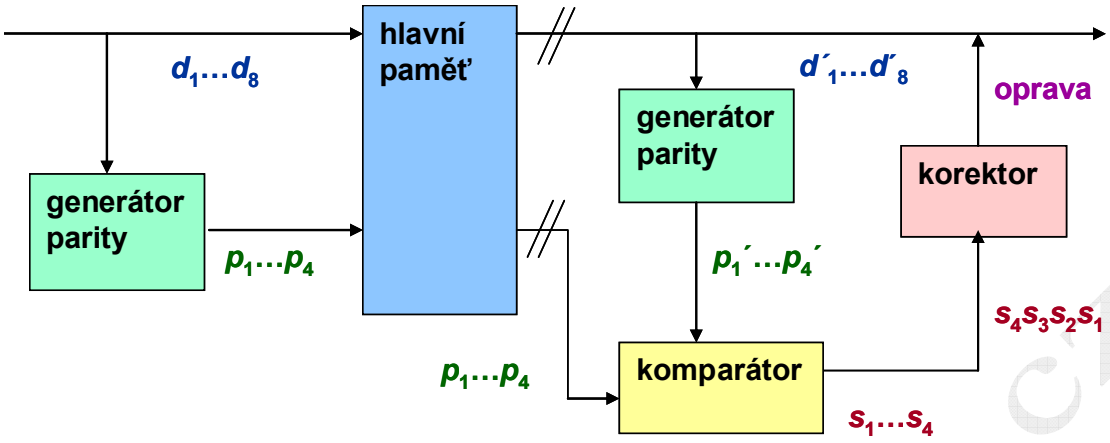
d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	p_1	p_2	p_3	p_4

- uspořádáno:

p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7	d_8

  - $p_1 = d_1 \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_7$,
  - $p_2 = d_1 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7$,
  - $p_3 = d_2 \oplus d_3 \oplus d_4 \oplus d_8$,
  - $p_4 = d_5 \oplus d_6 \oplus d_7 \oplus d_8$

- K datovým bitům d_1 ... d_8 generuje **generátor parity** paritní bity p_1 ... p_4



- **Komparátor** stanoví **příznaky** (**syndromy**):
  $s_1 = p_1 \oplus p_1'$ $s_3 = p_3 \oplus p_3'$
  $s_2 = p_2 \oplus p_2'$ $s_4 = p_4 \oplus p_4'$

- **Korektor** poté provede opravu chybného bitu, přičemž postupuje takto:
  - **všechny syndromy jsou 0** → žádná chyba detekována, korekce se neprovádí.
  - **pouze jeden syndrom je 1** → chyba je pouze v paritním bytu, oprava není nutná.
  - **více bitů má syndrom 1** → **mohlo dojít k chybě v datovém bitu, je třeba změnit na opačný bit s pořadím** $s_4\, s_3\, s_2\, s_1$.

  - **Při dvou chybách však opraví špatně !**

- **SEC DED** se získá **doplněním dalšího paritního bitu**
  - V praxi se dnes užívá nejčastěji SEC DED 64+8 nebo 32+7

- Nadbytečnost dat pro zabezpečení kódu je u současných počítačů pro uživatele zcela **NEVIDITELNÁ**
  - Uživatel ani autor programu se o ní nemusí starat → **vnitřní úroveň zabezpečení na úrovni kódu**
  - **Globální úroveň zabezpečení** = zálohování dat pro případ poruchy systému – uživatel i programátor musí zajistit

http://fit.pef.czu.cz