2003-10-10

Při ukládání, zpracování a přenosu dat je nutným požadavkem SPOLEHLIVOST.
Jeden z prostředků dosažení spolehlivosti je **REDUNDANCE** = nadbytečnost dat

Redundance na úrovni kódu
Kód = prosté zobrazení množiny kódovaných objektů do množiny kódových slov (posloupnosti 0 a 1)

$\{Oj\}$ - $\{Kj = k(Oj)\}$, $Oi \neq Oj \Rightarrow k(Oi) \neq k(Oj)$

HAMMINGOVA VZDÁLENOST KÓDOVÝCH SLOV – počet míst, na kterých se 0 a 1 liší.
$X = (x_1, x_2, \ldots, x_N)$ a $Y = (y_1, y_2, \ldots, y_N)$ – na kolika místech se slova liší

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

Počet míst, na kterých se kódová slova liší.
Poznámka: Hammingova vzdálenost tvoří z množiny kódových slov metrický prostor

1. $d(x,y) \geq 0$ a $d(x,y) = 0$ pouze když $x = y$
2. $d(x,y) = d(y,x)$ pro každé x a y
3. $d(x,y) \geq d(x,z) + d(z,y)$ pro každé x, y, z

Minimální Hammingova kódová vzdálenost kódu k:
$d_{\min} = \min_{x \neq y} d(x,y)$
Nejmenší vzdálenost dvou různých kódových slov

Zabezpečující kód – při změně bitu (z 0 na 1 nebo z 1 na 0) nemůže dojít k záměně za jiné kódové slovo (chyba v jednom bitu se detekuje)
$d_{\min} \geq 2$
Pokud dojde k chybě ve dvou a více bitech, může k záměně dojít.

Samoopravný kód – (single error corrected – 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 se opraví)
$d_{\min} \geq 3$
Pokud dojde 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.

Zabezpečující kód lze získat přidáním jednoho paritního bitu tak, aby počet jedniček byl

ü buď sudý = sudá parita (x1, …, xN) → (x1, …, xN, xN+1)

$$(\sum_{j=1}^{N+1} x_j) \bmod 2 = 0, x_{N+1} = x_1 \oplus x_2 \oplus \ldots \oplus x_N$$

ü nebo lichý = lichá parita (x1, …, xN) → (x1, …, xN, xN+1)

$$(\sum_{j=1}^{N+1} x_j) \bmod 2 = 0, x_{N+1} = x_1 \oplus x_2 \oplus \ldots \oplus x_N + 1.$$

FUNKCE Å = XOR = NOT
Exklusive OR – „vylučující nebo“ – antiekvivalence

Dává hodnotu 1 = true
Když jeden a jen jeden argument je 1 = true
Dává hodnotu 0 = false
Když jsou oba argumenty 0 = false nebo oba argumenty 1 = true

Pravdivostní tabulka funkce ⊕ = XOR = not

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	1

Odpovídá sčítání „modulu 2“ – zbytek součtu při dělení dvěma A ⊕ B = (A + B) mod2 (pouze representaci – vlevo jsou logické hodnoty, vpravo čísla)

Pro porovnání pravdivostní tabulky některých dalších logických funkcí:

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

Disjunkce
“OR”, ∨, +
nebo

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

Konjunkce
“AND”, ∧, -
a

A	B	A⇔B
0	0	1
0	1	0
1	0	0
1	1	1

Ekvivalence
⇔
tehdy a jen tehdy, právě když

REDUNDANCE PRO SAMOOPRAVNÝ (SEC) KÓD

Odvození: Binární kód délky m bitů, Hammingova minimální vzdálenost 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).
$2^m + (m+k) \cdot 2^m \leq 2^{(m+k)}$
$k + m + 1 \leq 2^k$

Tabulka nutné redundance 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

K 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

*d*1	*d*2	*d*3	*d*4	*d*5	*d*6	*d*7	*d*8	*p*1	*p*2	*p*3	*p*4

d = datové bity
p = paritní (pomocné) bity
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

$p1 = d1 \oplus d2 \oplus d4 \oplus d5 \oplus d7$
$p2 = d1 \oplus d3 \oplus d4 \oplus d6 \oplus d7$
$p4 = d2 \oplus d3 \oplus d4 \oplus d8$
$p4 = d5 \oplus d6 \oplus d7 \oplus d8$
Do paměti se uloží 12 bitů. Při čtení se k d1… d8 generují nové paritní bity p´1, p´2, p´3, p´4 a porovnají se:

$$\left.\begin{array}{l} s1 = p1 \oplus p'1 \\ s2 = p2 \oplus p'2 \\ s3 = p3 \oplus p'3 \end{array}\right\} \text{takzvané syndromy}$$

$s4 = p4 \oplus p'4$

Možné situace:

1. Všechny syndromy jsou 0 = vše je v pořádku.
2. Pouze jeden syndrom je 1 = chyba je pouze v paritním bytu oprava není nutná
3. Více bitů má syndrom 1 = Je třeba změnit opačný bit s pořadím s1 s2 s3 s4 – binárně

Při dvou chybách však opraví špatně.

SEC DED se získá doplněním dalšího paritního bitu.
V praxi se užívá SEC DED 64 + 8 nebo 32 + 7
Obecně pro opravu chyb je třeba aby $d_{min} = 2k + 1$
Jinými slovy: Automaticky lze vždy opravit současně $\frac{d_{min} - 1}{2}$ chyb. […] – celá část

Co mohou znamenat data uložená v počítači?

Informace (elementární datové typy)   Povely (instrukce programu)

Logické hodnoty   Znaky   Čísla

Pevná řádová čárka   Pohyblivá (plovoucí) řád. c.

Čísla nemají znamenat jen množství, velikost, ale i pořadí větší preference i označení kategorie do které entita patří.

POVELOVÁ DATA – pro instrukci počítače, tzv. **strojový kód**

Operační znak	**Adresová část**
Co se má udělat	S čím se to má udělat – 1 nebo více „operandů“

Poznámka: Kódování závisí na typu počítače.

Inforamační data:

Logická hodnota (Boolean):

- Ano (true) 1
- Ne (false) 0

K zobrazení stačí 1 bit – často se zobrazuje jediná v celém bytu nebo v celém slově (v adresovatalné jednotce). V 1 slově lze zobrazit víc logických hodnot.

ZNAK (character)

Řídící – konec zprávy, nová řádka…
Grafické – písmena, číslice, interpunkce, značky

Původně 7 bitů ASCII kód (128 možností – jen řídící znaky, anglická abeceda a základní interpunkční a jiné znaky)
Nyní převážně 8 bitů pro znak, včetně znaků národních abeced – malých i velkých (256 možností)

Normalizovány pro přenos dat jsou:

Latin1 – znaky „západoevropských“ abeced
Latin2 – znaky „východoevropských“ abeced – včetně naší
Latin3, Latin4… **-** další hláskové abecedy (řecká, azbuka, židovská…)

„Uvnitř“ výpočetních systémů se kódování může odlišovat:
Částečně – například kód firmy Microsoft
Zcela – například kód EBCDEC firmy IBM pro střediskové počítače
Existují i „zastaralé kódy češtiny“ – užívané před latin2, např. „kód bratří Kamenických“

Perspektivně asi UNICODE
2 byty pro jeden znak
2562 – 65.536 možností
Pokrývá s rezervou i slabinové a některé „slovové“ abecedy (3 japonské, čínskou) a další „exotické“ abecedy.
Dvojnásobná potřeba paměti.

ČÍSLA

Znaková (vnější) reprezentace jako znaky. Vhodné pro tisk, ne pro výpočet.

ü Pevná čárka:

Nezáporná čísla – dvojková soustava – ne záporná

Celá čísla (včetně záporných)

1. **Přímý kód**

   Znaménko — *Absolutní hodnota*
   1 bit — N-1 bitů
   + = 0 — nulu lze vyjádřit dvěma způsoby +0 = -0
   - = 1 — $x \in <-2^{N-1}-1, 2^{N-1}-1>$– nevhodné pro výpočet

2. **Kód s posunutou nulou**

   Zobrazí se číslo $x' = x + 2^{N-1}$ jako nezáporné číslo
   $x \in <-2^{N-1}, 2^{N-1}-1> \rightarrow x' \in <0, 2^{N}-1>$
   0 = 1000…0000 — nezáporná čísla mají nejvyšší bit 1
   n – 1 nul — záporná čísla mají nejvyšší bit 0

3. **Doplňkový kód** – převažuje

   Zn: + = 0, - = 1 — Číslo nebo jako doplněk
   1 bit — N - 1 bitů
   $d(x) = x$, pokud $0 \leq x \leq 2^{N-1}-1$ — a) jako 0 - x
   $d(x) = x+2^{N}$, pokud $-2^{N-1} \leq x < 0$ — b) $0 \rightarrow 1$, $1 \rightarrow 0$ a +1
   $<-2^{N-1}, 2^{N-1}-1> \rightarrow <0, 2^{N}-1>$
   d — vhodné pro aritmetiku

   Integer N=16, $x \in <-32768, 32767>$,
   Longint N=32, $x \in <-2^{31}, 2^{31}-1>$.

ü Pohyblivá (plovoucí) čárka

$X = m \,.\, z^{e}$

z – základ u dekadických logaritmů 10, zde 2
m – mantisa
e – exponent

Mantisa je normalizovaná, je-li to číslo intervalu $<z^{-1}, 1)$ při z = 2 tady $m \in <1/2, 1)$.

Formát IEEE čísla Real

4 byty = 32 bitů

z	e	m
0 = +	8 bitů	23 bitů
1 = -	posunutá 0	nejvyšší je vždy 1, tedy se vynechá