ZOBRAZOVANIE ČÍSEL V PEVNEJ RÁDOVEJ ČIARKE

Informatika v počítači – to je predovšetkým práca s číslami. Aritmetické operácie s nimi nie sú v princípe zložité, no už jednoduché „ručné" spočítanie dvoch čísel robí študentom problémy: bez použitia kalkulátora často robia chyby. S odpočítavaním je to ešte horšie, niekedy nedokážu nájsť v primeranom čase správny výsledok takej operácie. Preto tu chceme poskytnúť vyučujúcim i študentom niekoľko názorných príkladov, ako to robí počítač, v ktorom sa čísla (okrem iných zobrazení) zapisujú vo formáte, ktorý označujeme ako **pevná rádová čiarka**. Článok poskytuje dosť námetov na praktické precvičovanie problematiky
.

Je známe, že v počítači sa nerobí operácia odčítania samostatnou odčítačkou, ale volí sa také zobrazenie čísel, ktoré umožní použiť bežnú sčítačku aj na odčítanie. Tu uvedieme tri spôsoby zobrazenia, pre ktoré sa zaužívali názvy **priamy, doplnkový a inverzný kód**.

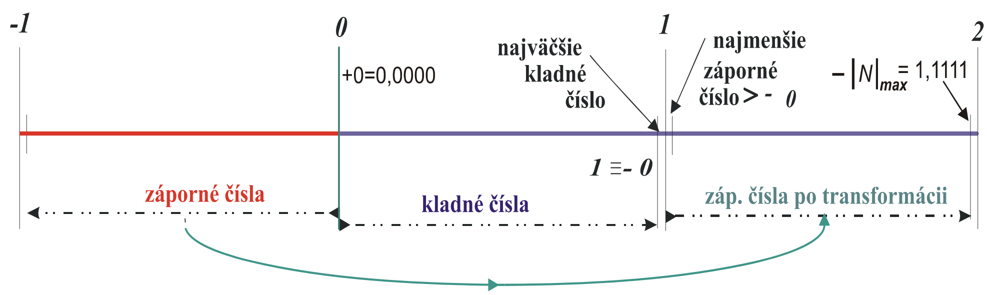
1. Priamy kód

Ide o zobrazenie informačného slova typu číslo, v pevnej rádovej čiarke, označme ho x, pre ktoré platí obmedzenie

$$|x| < 1 \qquad (1)$$

$$x_{pr} = \begin{cases} x, & \text{pre } x \geq 0 \\ 1 - x, & \text{pre } x \leq 0 \end{cases} \qquad (2)$$

Z uvedeného vyplýva, že číslo, napr. $+\,5/16$, bude v dvojkovom zápise 0, 0101, a číslo – $5/16$ bude vyjadrené 1, 0101. Prvý bit vľavo od rádovej čiarky sa označuje ako znamienkový , a ako vidno, kladné číslo má v ňom nulu, záporné jednotku, bity mantisy sú zhodné. Takéto zobrazenie čísel priamym kódom sa používa na ukladanie čísel do pamäti. Presvedčte sa na číselnej osi *(obr. 1)*, že zobrazenie jednotky je zhodné s obrazom zápornej nuly.



Obr. 1: Číselná os k priamemu kódu

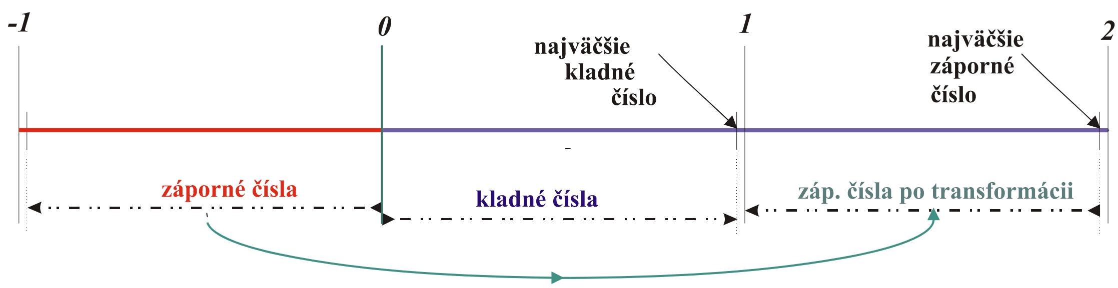
2. Doplnkový kód

Ide opäť o zobrazenie informačného slova typu číslo v číselnej sústave o základe *10*, v pevnej rádovej čiarke, označme ho x, pre ktoré platí obmedzenie

$$|x| < 1 \qquad (3)$$

$$x_{dop} = \begin{cases} x, & \text{pre } x \geq 0 \\ 10 + x, & \text{pre } x < 0 \end{cases} \qquad (4)$$

Uvedené vzťahy vedú na zobrazenie číselnej osi podľa obr. 2. Doplnkový kód kladného čísla x je opäť rovný tomu kladnému číslu x. Doplnok záporného čísla dostaneme tak, že do znamienkového bitu zapíšeme hodnotu *1*, na všetkých ďalších nahradíme jednotky nulami a nuly jednotkami a k najnižšiemu rádu pripočítame jednotku. Je to priamy dôsledok vzťahu (4.) a zároveň ukazuje, odkiaľ kód dostal svoj názov: obraz záporného čísla je totiž doplnkom do základu sústavy (teda tu do dvojky).



Obr.2: Číselná os k doplnkovému kódu

Ako vidno, nula je vždy kladná +0,00000, maximálne záporné číslo je 1,00000 a nemá kladný ekvivalent. Dá sa dokázať, že súčet dvoch čísel v doplnkovom kóde dáva opäť doplnok súčtu: $x_{dop} + y_{dop} = (x + y)_{dop}$.

Príklad 1

$$\begin{array}{rl} x = & 0{,}1101 \\ y = & 0{,}0001 \\ \hline x+y = & 0{,}1110 \end{array} \qquad \begin{array}{rl} x_{dop} = & 0{,}1101 \\ y_{dop} = & 0{,}0001 \\ \hline (x+y)_{dop} = & 0{,}1110 \end{array}$$

Príklad 2

$$\begin{array}{rl} x = & +0{,}1101 \\ y = & -0{,}0001 \\ \hline x+y = & 0{,}1100 \end{array} \qquad \begin{array}{rl} x_{dop} = & 0{,}1101 \\ y_{dop} = & 1{,}1111 \\ \hline (x+y)_{dop} = & 10{,}1100 \end{array}$$

neuvažuje sa

Príklad 3

$$\begin{array}{rlrl} x = & -0{,}1101 & x_{dop} = & 1{,}0011 \\ y = & +0{,}0001 & y_{dop} = & 0{,}0001 \\ \hline x+y = & -0{,}1100 & (x+y)_{dop} = & 1{,}0100 \end{array}$$

Príklad 4

$$\begin{array}{rlrl} x = & -0{,}1101 & x_{dop} = & 1{,}0011 \\ y = & -0{,}0001 & y_{dop} = & 1{,}1111 \\ \hline x+y = & -0{,}1110 & (x+y)_{dop} = & 11{,}0010 \end{array}$$

neuvažuje sa

Príklad 5

$$\begin{array}{rlrl} x = & -0{,}1101 & x_{dop} = & 1{,}0011 \\ y = & -0{,}0111 & y_{dop} = & 1{,}1001 \\ \hline x+y = & -1{,}0100 & (x+y)_{dop} = & 10{,}1100 \end{array}$$

indikuje, že ide o kladný výsledok

Príklad 5 ukazuje prípad vzniku chybného výsledku, ktorý by mal byť isto záporný, lebo oba sčítance sú záporné. Táto chyba vznikla pretečením číselného rozsahu bitov mantisy do znamienkového bitu. Pripomeňme, že ak platí vzťah (3) pre sčítance, musí platiť aj pre ich súčet. (V tomto príklade sa hľadal súčet čísel –13/16 a –7/16 čo je –20/16 a je zrejmé že na to nestačia 4 bity mantisy, ale musí ich byť aspoň päť. Preto vzniklo pretečenie a výsledok je nepoužiteľný).

3. Inverzný kód

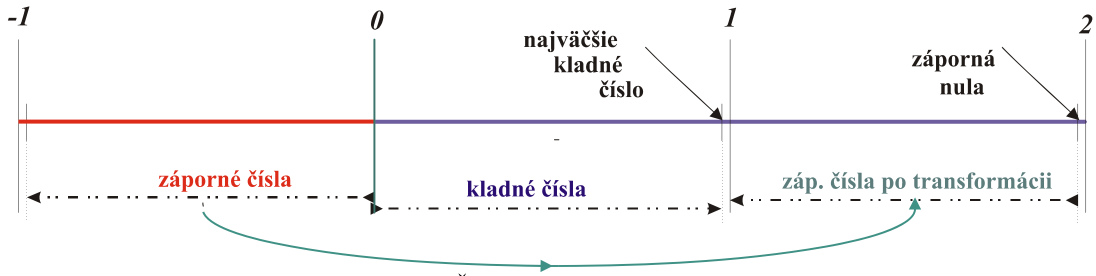
Rovnako ako vyššie, ide o zobrazenie informačného slova typu číslo v číselnej sústave o základe *10*, v pevnej rádovej čiarke, označme ho x, pre ktoré platí obmedzenie

$$|x| < 1 \qquad (5)$$

$$x_{inv} = \begin{cases} x, & \text{pre } x \geq 0 \\ 10 + x - 1.\ 10^{-n}, & \text{pre } x \leq 0 \end{cases} \qquad (6)$$

Ak porovnáte vzťahy (4 a 6) ľahko zistíte, že inverzný kód pre záporné x dáva číslo, ktoré je o jednotku najnižšieho rádu menšie než u doplnkového kódu: nie je to doplnok do základu ústavy, ale do základu sústavy zmenšeného o jednotku najnižšieho (*n-tého*) rádu. Pre dvojkovú sústavu ide o doplnok do čísla $\boldsymbol{10 - 1.\ 10^{-n} = 1{,}111...1}$.

Uvedené vzťahy vedú na zobrazenie číselnej osi podľa *obr. 3.* Inverzný kód kladného čísla x je opäť rovný tomu kladnému číslu x. Inverzný kód záporného čísla dostaneme tak, že vo všetkých bitoch nahradíme jednotky nulami a nuly jednotkami. Je to priamy dôsledok vzťahu (6) a zároveň ukazuje, odkiaľ kód dostal svoj názov: obraz záporného čísla vznikne totiž inverziou pôvodných hodnôt v jednotlivých bitoch.



Obr.3: Číselná os k inverznému kódu

Kladná nula dáva obraz 0,00000, záporná nula 1,11111. Pre operáciu sčítania v inverznom kóde musíme uvažovať aj takýto možný prípad:

$$\begin{array}{rr} & 10 + \boldsymbol{x} - 1.\ 10^{-n} \\ + & 10 + \boldsymbol{y} - 1.\ 10^{-n} \\ \hline & 100 + (\boldsymbol{x}+\boldsymbol{y}) - 2\ .10^{-n} \end{array} \qquad (7)$$

Ľahko poznáme, že vzťah (7) nie je zhodný so vzťahom (6) definujúcim inverzný kód, ale že vznikol iný kód, ktorý predstavuje o dve jednotky menšie číslo než má mať obraz inverzného kódu. Preto musíme v takomto prípade robiť korekciu, t.j. pripočítať k výsledku jednu jednotku najnižšieho rádu, aby aj pre vyjadrenie súčtu bol zachovaný rovnaký definičný vzťah. Takáto korekcia sa robí formou tzv. **kruhového prenosu,** keď sa využije fakt, že v znamienkovom bite vznikol prenos do vyššieho rádu, ktorý možno pripočítať k najnižšiemu rádu.

Príklad 6

$$\begin{array}{rl@{\qquad}rl} x = & 0,1101 & x_{inv} = & 0,1101 \\ y = & 0,0001 & y_{inv} = & 0,0001 \\ \hline x+y = & 0,1110 & (x+y)_{inv} = & 0,1110 \end{array}$$

Príklad 7

$$\begin{array}{rl@{\qquad}rl@{\qquad}rl} x = & -\ 0,1101 & x_{inv} = & 1,0010 & & 1,1111 (\text{zákl}-10^{-4}) \\ y = & +0,0001 & y_{inv} = & 0,0001 & - & 0,1101 \quad (x) \\ \hline x+y = & -\ 0,1100 & (x+y)_{inv} = & 1,0011 & & 1,0010 \quad (x_{inv}) \\ & & & & & \textit{inverzia} \end{array}$$

Príklad 8

$$\begin{array}{rl@{\qquad}rl} x = & +\ 0,1101 & x_{inv} = & 0,1101 \\ y = & -\ 0,0001 & y_{inv} = & 1,1110 \\ \hline x+y = & 0,1100 & (x+y)_{inv} = & 10,1011 \\ & & \textit{kruhový prenos} & \longrightarrow 1 \\ \hline & & (x+y)_{inv} = & 0,1100 \end{array}$$

Príklad 9

$$\begin{array}{rl@{\qquad}rl} x = & -\ 0,1101 & x_{inv} = & 1,0010 \\ y = & -\ 0,0001 & y_{inv} = & 1,1110 \\ \hline x+y = & -\ 0,1100 & (x+y)_{inv} = & 11,0000 \\ & & \textit{kruhový prenos} & \longrightarrow 1 \\ \hline & & (x+y)_{inv} = & 1,0001 \end{array}$$

4. Modifikované doplnky

Všimnime si teraz doplnkový kód. Povedali sme, že nula je tu vždy kladná: 0,000...00 a záporná nula nie je definovaná. Vzhľadom na to a tiež ak vieme, že je jediný bit pre znamienko, je celkový rozsah zobrazení záporných čísel väčší než rozsah kladných čísel a to o jednotku najnižšieho rádu. Maximálne záporné číslo 1,00...00 totiž nemá kladný ekvivalent : jeho pravý dvojkový doplnok je opäť 1,00...00, t.j. zase záporné číslo alebo číslo kladné, v ktorom sa prekročil (pretiekol) rozsah, vymedzený pre kladné čísla.

Rozsah môže pretiecť tiež pri spočítaní dvojice kladných alebo záporných čísel a prejaví sa to tým, že pri spočítaní dvojice kladných čísel dostávame v znamienkovom bite jedničku (čo je znak záporného čísla) a naopak, pri spočítaní dvoch záporných čísel nulu, (čo je znak kladného čísla). Ak chceme mať korektný výsledok, museli by sme si pamätať znamienka sčítancov a porovnávať ich so znamienkom výsledku. Lepšie je však pracovať s tzv. **modifikovanými doplnkami.**
Hlavná myšlienka spočíva v tom, že za základ číselnej sústavy zoberieme jej vyššiu mocninu: namiesto *10* (dvojka) budeme pracovať so základom *100* (štvorka). Ak zobrazujeme v súlade s predpokladom číslo $x < 1$, potom doplnok sa vytvára do štvorky a prestavuje dva bity. Kladné číslo má potom oba tieto bity nulové (00,...) a záporné jednotkové (11,...). Prípad v ktorom v znamienkových bitoch vznikne podoba (01,...) alebo (10,...) indikuje **stav pretečenia.** Znamienko v tomto prípade určuje vyšší znamienkový rád, a nižší bit v tej dvojici sa označuje aj ako rád pretečenia.
V uvažovanom prípade potom kódové obrazy vytvoríme podľa vzťahov (8) pre doplnkový kód a (9) pre inverzný kód :

$$x_{dop} = \begin{cases} x, & \text{pre } x \geq 0 \\ 100 + x, & \text{pre } x < 0 \end{cases} \qquad (8)$$

$$x_{inv} = \begin{cases} x, & \text{pre } x \geq 0 \\ 100 + x - 1.\,10^{-n}, & \text{pre } x \leq 0 \end{cases} \qquad (9)$$

Príklad 10

$$\begin{array}{rlrl} x &= -0{,}1101 & x_{dop} &= 11{,}0011 \\ y &= -0{,}0001 & y_{dop} &= 11{,}1111 \\ \hline x+y &= -0{,}1110 & (x+y)_{dop} &= 11{,}0010 \end{array}$$

neuvažuje sa

Príklad 11

$$\begin{array}{rlrl} x &= -0{,}1101 & x_{inv} &= 11{,}0010 \\ y &= -0{,}0001 & y_{inv} &= 11{,}1110 \\ \hline x+y &= -0{,}1110 & (x+y)_{inv} &= 111{,}0000 \\ & & \textit{kruhový prenos} & +1 \\ \hline & & (x+y)_{inv} &= 11{,}0001 \end{array}$$

5. Desiatkový a deviatkový doplnok

V desiatkovej sústave sú operácie s číslami celkom analogické. Pre doplnkový kód platia vzťahy (3) a (4) s tým, že sa za základ sústavy volí desiatková desiatka. Kódový obraz záporného čísla je teda **desiatkový** doplnok.

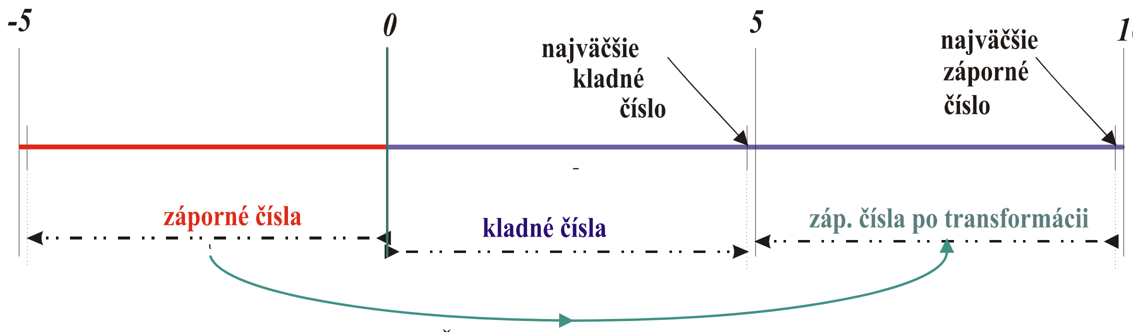
Príklad 12

$x = +0{,}4257$	$x_{dop} = 0{,}4257$	$10{,}0000$ (základ)
$y = -0{,}0316$	$y_{dop} = 9{,}9684$	$-0{,}0316$ (y)
$x+y = +0{,}3941$	$(x+y)_{dop} = 10{,}3941$	$9{,}9684$ (y_{dop})
	neuvažuje sa	*desiatkový doplnok*

Všimnite si, že kladné číslo má v znamienkovom mieste nulu, záporné deviatku. V desiatkovej sústave sa však používa desať číslic 0, 1, 2, ...9, čo umožňuje uvoľniť obmedzenie podľa vzťahu (3) takto:

$$|x| < 5 \qquad (10)$$

čo rozdeľuje číselnú os na dva intervaly: $\langle 0, 5)$ pre kladné čísla a $\langle 5,10)$ pre čísla záporné, pozri obr. 4. Nula je vždy kladná (+0,00...00). Najväčšie záporné číslo je 5,00...00 a nemá kladný ekvivalent.



Obr. 4: Číselná os pre desiatkové doplnky

Príklad 13

$x = +4{,}4257$	$x_{dop} = 4{,}4257$	$10{,}0000$ (základ)
$y = -2{,}6966$	$y_{dop} = 7{,}3034$	$-2{,}6966$ (y)
$x+y = +1{,}7291$	$(x+y)_{dop} = 11{,}7291$	$7{,}3034$ (y_{dop})
	neuvažuje sa	

Vo všeobecnosti platí, že číslice **0,1,2,3,4** zapísané v znamienkovom mieste indikujú číslo kladné, číslice **5,6,7,8,9** znamenajú že ide o číslo záporné, vyjadrené desiatkovým doplnkom. Ale uvedomme si tiež, že v týchto úvahách sa ukazuje, že číslica zapísaná v znamienkovom mieste (prvé vľavo od rádovej čiarky), v zmysle vzťahu (10) už nenesie len informáciu o znamienku daného čísla, ale spoluvytvára celú jeho číselnú hodnotu. **To je dôležité pre spätný prevod do priameho kódu.**

Príklad 14

$x = -4{,}265$	$x_{dop} = 5{,}745$	$10{,}000$ (základ)	$10{,}000$ (základ)
$y = -3{,}431$	$y_{dop} = 6{,}569$	$-3{,}431$ (y)	$-4{,}265$ (x)
$x+y = -7{,}696$	$(x+y)_{dop} = 12{,}314$	$6{,}569$ (y_{dop})	$5{,}745$ (x_{dop})

indikuje kladné číslo, takže ide o ***chybu pretečením.***

Inverzný kód v desiatkovej sústave nazývame **deviatkovým doplnkom**. Vyplýva to rovno zo vzťahu (9) : $10 - 1.\,10^{-n} = 9{,}999...99.$ Tu je skrytý postup tvorby kódových obrazov.

Príklad 15

$$\begin{array}{lll}
x = -3{,}4253 & x_{inv} = 6{,}5746 & 9{,}9999 \quad (\text{základ}-10^{-4}) \\
y = +1{,}1926 & y_{inv} = 1{,}1926 & -3{,}4253 \quad (x) \\
\hline
x+y = -2{,}2327 & (x+y)_{inv} = 7{,}7572 & 6{,}5746 \quad (x_{inv}) \\
 & \uparrow \textit{výsledok} & \textit{deviatkový doplnok}
\end{array}$$

Výsledok je – ako vidno – záporný a je to teda deviatkový doplnok záporného čísla, ktoré je súčtom oboch pôvodných čísel vyjadrených v inverznom kóde. Ak urobíte spätný deviatkový doplnok z tohoto doplnku, dostanete „viditeľnú" kontrolu toho, že nedošlo k žiadnej chybe:

$$-9{,}9999 + 7{,}7572 = -2{,}2327.$$

Pripomeňme, že vo všeobecnosti treba robiť korekciu pri súčte v tom istom duchu ako pri sústave dvojkovej a uplatniť **kruhový prenos.**

Príklad 16

$$\begin{array}{lll}
x = -4{,}3425 & x_{inv} = 5{,}6574 & 9{,}9999 \quad (\text{základ}-10^{-4}) \\
y = -3{,}4192 & y_{inv} = 6{,}5807 & -3{,}4192 \quad (y) \\
\hline
x+y = -7{,}7617 & (x+y)_{inv} = 12{,}2381 & 6{,}5807 \quad (y_{inv}) \\
 & \phantom{(x+y)_{inv} = 1}+1 & \\
 & \phantom{(x+y)_{inv} = 1}2{,}2382 &
\end{array}$$

*indikuje kladné číslo, takže ide o **chybu pretečením.***

Príklad 17

$$\begin{array}{lll}
x = -1{,}5412 & x_{inv} = 8{,}4587 & 9{,}9999 \quad (\text{základ}-10^{-4}) \\
y = -1{,}0374 & y_{inv} = 8{,}9625 & -1{,}5412 \quad (x) \\
\hline
x+y = -2{,}5789 & (x+y)_{inv} = 17{,}4212 & 8{,}4587 \quad (x_{inv}) \\
 & \phantom{(x+y)_{inv} = 1}+1 & \\
 & \phantom{(x+y)_{inv} = 1}7{,}4213 \leftarrow & \textit{znamienko aj výsledok je správny}
\end{array}$$

Uvedené pojmy sú jednoduché, sú vhodnou témou na cvičenia s číslami v pevnej rádovej čiarke a v podobnom duchu pripravíme i témy s číslami v pohyblivej rádovej čiarke. Teraz však treba objasniť i prácu s číslami x, ktoré nespĺňajú podmienku $|x|<1$ danú napr. vzťahom (1), ale sú v absolútnej hodnote väčšie ako 1. Je celkom prirodzené, že i takéto čísla musia byť počítačom spracovateľné aj v pevnej rádovej čiarke.

Problém nie je nijak zložitý ak zavedieme pojem *merná jednotka.* Uvažujme napr. číslo $x = +13/16$. Čitateľ tohto zlomku, t.j. číslo *13* je väčšie ako *1* a výraz môžeme čítať ako *trinásť šestnástin.* Menovateľ *16* je tiež väčší ako *1*, ale hodnota zlomku *13/16* je určite menšia ako *1*, lebo menovateľ je väčší ako čitateľ. Z toho plynie, že číslo *13* bolo transformované do podoby vyhovujúcej vzťahu (1) pre zobrazenie v pevnej rádovej čiarke a mernou jednotkou sa stali *šestnástiny.* Dvojkovo sa uvažované $x = +13/16$ zapíše napríklad v priamom kóde ako 0,1101. Uvedomte si, že váhy v bitoch napravo od rádovej čiarky postupne vyjadrujú *poloviny, štvrtiny, osminy, šestnástiny...,* t.j. 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, ... , 2^{-k} , kde k je počet bitov napravo od rádovej čiarky a predstavuje mernú jednotku.

V praktických úlohách treba voliť takú hodnotu k, ktorá by minimalizovala možnosť vzniku pretečenia pri realizácii operácií, v spracovaní počítačom sa automaticky volí tá merná jednotka , ktorá prislúcha poslednému pravému bitu v slove.

Objasnime ešte pojem mernej jednotky pre desiatkovú sústavu. Nech číslo $x = 653$. Číslo samo o sebe je bezrozmerné, ale predpokladajme, že predstavuje finančnú hodnotu vyjadrenú v korunách. (Mohli by sme tiež hovoriť, že ide o *653* jabĺk, *653* rokov, *653* litrov, atď., ale z hľadiska mernej jednotky sú peňažné názvy najnázornejšie.)

Môžeme zapísať takéto vyjadrenia: merná jednotka

653	korún	jednotky
65,3	desaťkorún	desatiny
6,53	stokorún	stotiny
0,653	tisíckorún	tisíciny
0,0653	desaťtisíckorún	desaťtisíciny
0,00653	stotisíckorún	stotisíciny

Číselný zápis je vo všetkých riadkoch rovnaký, ale formálny zápis vyhovujúci podmienke vzťahu (1) je vhodný až od štvrtého riadku zoznamu. Teda mernou jednotkou musia byť prinajmenšom *tisíciny* alebo ešte menšie hodnoty, aby bolo možné vyjadriť x v pevnej rádovej čiarke v niektorom z preberaných kódov. Procesu prepisu čísla do podoby z vhodnou mernou jednotkou sa hovorí **zmena mierky.** V praktickej podobe tomu zodpovedá *posuv čísla doprava* o potrebný počet miest (v registri, v ktorom je číslo uložené). Po vykonaní operácie sa treba vrátiť k pôvodnej mierke.

Príklad 18

Vypočítajte, koľko rokov má človek v roku 2005, keď sa narodil v roku 1962.

Označme ***x´= 2005***, ***y´= – 1962,*** zmeňme mierku posuvom čísel o 4 miesta doprava a použime napríklad doplnkový kód. Potom

$x = +\,0{,}2005$	$x_{dop} = 0{,}2005$	$10{,}0000$ (základ)
$y = -0{,}1962$	$y_{dop} = 9{,}8038$	$-\,0{,}1962$ (y)
$x+y = +\,0{,}0043$	$(x+y)_{dop} = 10{,}0043$	$9{,}8038$ (y_{dop})

neuvažuje sa

Výsledok operácie je v prostrednom stĺpci, ide o číslo kladné a po návrate k pôvodnej mierke možno odpovedať: v roku 2005 dosiahol človek z príkladu 43 rokov.

Prof. Ing. Ján KOLENIČKA, PhD.
Ing. Jarmila ŠKRINÁROVÁ, PhD.
Katedra informatiky FPV UMB
Banská Bystrica
kolenick@fpv.umb.sk
skrinar@fpv.umb.sk