

Data v počítači

■ Informační data (*elementární datové typy*)

■ Logické hodnoty

■ Znak

■ Číslo

■ v pevné řádové čárce (*celá čísla*)

■ v pohyblivé (plovoucí) řád. čárce (*reálná čísla*)

■ Povelová data (*instrukce programu*)

Informační data

Logické hodnoty

■ Logická hodnota (*boolean*)

■ ANO (*true*) 1

■ NE (*false*) 0

■ K zobrazení stačí 1 bit

■ Často se zobrazuje jediná hodnota v celém bytu nebo v celém slově (v adresovatelné jednotce).

■ V 1 slově lze zobrazit více logických hodnot

Znakové hodnoty

■ ZNAK (*Character*)

■ řídicí (konec zprávy, nová řádka, ...)

■ grafický (písmeno, číslice, interpunkce, značky)

■ Reprezentace

■ Původně 7 bitů ASCII kód (128 možností – jen řídicí znaky, anglická abeceda a základní interpunkční a jiné znaky)

■ Nyní převážně 8 bitů ASCII kód, včetně znaků národních abeced – malých i velkých (256 možností)

■ Normalizovány pro přenos dat jsou:

■ LATIN1 (znaky „západoevropských“ abeced)

■ LATIN2 (znaky „východoevropských“ abeced – včetně naší české abecedy)

■ LATIN3, LATIN4 ... další hláskové abecedy (řecká, azbuka, židovská, arabská, ...)

– je nutné přepínat mezi kódy

■ Nyní se prosazuje **UNICODE**

- **2 byty pro jeden znak** $256^2 = 65\,536$ možností
- pokrývá s rezervou i slabikové a některé „slovové“ abecedy (3 abecedy japonské, čínskou, ...) a další „exotické“ abecedy
- *dvojnásobná potřeba paměti*

■ Tabulka ASCII – LATIN2

Regular ASCII Chart (character codes 0 - 127)									
000	(nul)	016	► (dle)	032	sp	048	0	064	Ø
001	☉ (soh)	017	◄ (dc1)	033	!	049	1	065	À
002	⊕ (stx)	018	↑ (dc2)	034	"	050	2	066	B
003	⊖ (etx)	019	!! (dc3)	035	#	051	3	067	C
004	+ (eot)	020	¶ (dc4)	036	\$	052	4	068	D
005	⊞ (enq)	021	§ (nak)	037	%	053	5	069	E
006	⊠ (ack)	022	— (syn)	038	&	054	6	070	F
007	• (bel)	023	↑ (etb)	039	'	055	7	071	G
008	▣ (bs)	024	↑ (can)	040	(	056	8	072	H
009	(tab)	025	↓ (em)	041	)	057	9	073	I
010	(lf)	026	(eof)	042	*	058	:	074	J
011	♂ (vt)	027	← (esc)	043	+	059	;	075	K
012	* (np)	028	L (fs)	044	,	060	<	076	L
013	(cr)	029	→ (gs)	045	-	061	=	077	M
014	♂ (so)	030	▲ (rs)	046	.	062	>	078	N
015	⊙ (si)	031	▼ (us)	047	/	063	?	079	O
								095	—
								111	o
								112	p
								113	q
								114	r
								115	s
								116	t
								117	u
								118	v
								119	w
								120	x
								121	y
								122	z
								123	{
								124	
								125	}
								126	~
								127	□

Extended ASCII Chart (character codes 128 - 255)															
128	Ç	143	Č	158	×	172	Č	186		200	Ł	214	í	228	ñ
129	ù	144	É	159	č	173	š	187	]]	201	ł	215	î	229	ň
130	é	145	Ĺ	160	á	174	«	188	]]	202	Ł	216	ë	230	š
131	â	146	Í	161	í	175	»	189	Ž	203	ł	217	ĵ	231	š
132	ä	147	ô	162	ó	176	⋮	190	ž	204	Ł	218	ŕ	232	ř
133	ũ	148	ö	163	ú	177	⋮	191	ŕ	205	Ł	219	ŕ	233	Ů
134	č	149	Ľ	164	Ā	178	⋮	192	ŕ	206	Ł	220	ŕ	234	ř
135	ç	150	Ĺ	165	ā	179	⋮	193	ŕ	207	Ł	221	ŕ	235	Ů
136	ı	151	Š	166	ž	180	⋮	194	ŕ	208	Ł	222	Ů	236	ý
137	ë	152	š	167	ž	181	Ā	195	ŕ	209	Ł	223	ŕ	237	Ý
138	Ö	153	Ö	168	Ē	182	Ā	196	—	210	Ď	224	Ó	238	ť
139	ö	154	Ů	169	ē	183	Ē	197	+	211	Ě	225	ß	239	’
140	í	155	Ť	170	¬	184	Š	198	Ā	212	ď	226	Ô	240	—
141	ž	156	ť	171	ž	185		199	ä	213	Ň	227	Ň	241	~
142	Ā	157	Ľ											255	■

Číselné hodnoty

■ ČÍSLA

- **Znaková** (vnější) **reprezentace** – číslíce jako znaky.
 - *Vhodné pro zobrazování a tisk, ne pro výpočet.*
- **Pevná řádová čárka** (celá čísla – kladná i záporná)
 - **Přímý kód**
 - **Kód s posunutou nulou**
 - **Doplňkový kód**
- **Pohyblivá (plovoucí) řádová čárka** (reálná čísla)
 - **IEEE formát čísla** - nejčastější

Přímý kód

■ Přímý kód - definice

- zobrazení množiny celých čísel x z intervalu $\langle -2^{n-1}, 2^{n-1} - 1 \rangle$ na množinu binárních kódových slov x' délky n .
- Pokud binární kódová slova x' vyjádříme celým číslem, lze toto zobrazení popsat jako:

$$\begin{aligned} \text{■ } x' &= x & \text{pokud } 0 \leq x \leq 2^{n-1} - 1 \\ \text{■ } x' &= 2^{n-1} - x & \text{pokud } -2^{n-1} + 1 \leq x \leq 0 \end{aligned}$$

- **Příklad:** (4-bitový kód, $n = 4$)

$$\begin{aligned} \text{■ } x = 5 & & x' = 5 & & 0101 \\ \text{■ } x = -5 & & x' = 2^{4-1} - 5 = 13 & & 1101 \end{aligned}$$

■ Přímý kód - vlastnosti

- první bit zleva se nazývá **znaménkový bit**
 - 0 ~ kladné číslo
 - 1 ~ záporné číslo
- zbývající bity obsahují **absolutní hodnotu** čísla x
- *Nula může být vyjádřena dvěma kódovými slovy:*
 - 000.....0 **kladná nula**,
 - 100.....0 **záporná nula**
 - **nevhodné pro výpočty**

Kód s posunutou nulou

■ Kód s posunutou nulou - definice

- zobrazení množiny celých čísel x z intervalu $\langle -2^{n-1}, 2^{n-1} - 1 \rangle$ na množinu binárních kódových slov x' délky n .
- Pokud binární kódová slova x' vyjádříme celým číslem, lze toto zobrazení popsat jako:

$$\text{■ } x' = 2^{n-1} + x \quad \text{pokud} \quad -2^{n-1} \leq x \leq 2^{n-1} - 1$$

- **Příklad:** (4-bitový kód, $n = 4$)

$$\begin{aligned} \text{■ } x = 5 & & x' = 2^{4-1} + 5 = 13 & & 1101 \\ \text{■ } x = -5 & & x' = 8 + -5 = 3 & & 0011 \end{aligned}$$

■ Kód s posunutou nulou - vlastnosti

- **Kladná čísla** nula mají v prvním bitu zleva **1**.
- **Záporná čísla** mají v prvním bitu zleva **0**.
- **Nula** je zobrazena bitovým řetězcem **100.....0**
- Používá se při zobrazení exponentu u čísel s pohyblivou řádovou čárkou

Doplňkový kód

Doplňkový kód - definice

- zobrazení množiny celých čísel x z intervalu $\langle -2^{n-1}, 2^{n-1} - 1 \rangle$ na množinu binárních kódových slov x' délky n .
- Pokud binární kódová slova x' vyjádříme celým číslem, lze toto zobrazení popsat jako:

$$\begin{aligned} \text{■ } x' &= x & \text{pokud } 0 \leq x \leq 2^{n-1} - 1 \\ \text{■ } x' &= 2^n + x & \text{pokud } -2^{n-1} \leq x \leq 0 \end{aligned}$$

- **Příklad:** (4-bitový kód, $n = 4$)

$$\begin{aligned} \text{■ } x = 5 & & x' = 5 & & 0101 \\ \text{■ } x = -5 & & x' = 2^4 + -5 = 11 & & 1011 \end{aligned}$$

Doplňkový kód – vlastnosti

- U **nezáporného čísla** je hodnota prvního bitu zleva **0**.
- U **záporného čísla** je hodnota prvního bitu zleva **1**.
- Kód vhodný pro aritmetické operace.

Doplňek - $d(x)$

- $x + d(x) = 0$ $d(x) = -x$
- **doplňek $d(x)$ k číslu x získáme:**
 - v binárním zápisu x zaměníme všechny **0 za 1** a všechny **1 za 0**
 - **přičteme binárně 1** → výsledek je $d(x)$

Sčítání čísel v doplňkovém kódu

- Jsou dána dvě čísla x a y v n -bitovém doplňkovém kódu. Kódová slova reprezentující čísla x a y můžeme zapsat jako řetězce binárních hodnot:
 - $x = x_{n-1} x_{n-2} \dots x_1 x_0$ $y = y_{n-1} y_{n-2} \dots y_1 y_0$
- K těmto číslům stanovíme **řetězec $c = c_n c_{n-1} c_{n-2} \dots c_1 c_0$** (*carry* = přenos), podle těchto pravidel:
 - $c_0 = 0$,
 - $c_{i+1} = 1$, pokud alespoň dvě z hodnot x_i, y_i a c_i jsou rovny 1. V opačném případě je hodnota $c_{i+1} = 0$,
- Dále stanovíme **řetězec $s = s_{n-1} s_{n-2} \dots s_1 s_0$** (součet), tímto způsobem: $s_i = x_i \oplus y_i \oplus c_i$,
- Pokud je hodnota $v = c_n \oplus c_{n-1} = 1$, došlo k tzv. **přetečení** a součet čísel x a y nelze stanovit. Pokud je hodnota $v = c_n \oplus c_{n-1} = 0$, potom řetězec s vyjadřuje součet $x + y$.

■ Příklad:

- V 6-bitovém doplňkovém kódu jsou dána čísla $x = 9$ a $y = -18$. Stanovte součet těchto čísel.

$$\begin{array}{rcl} 001001 & \text{hodnota } x & \\ + 101110 & \text{hodnota } y & \\ \hline \textcircled{00}10000 & \text{přenos } (c_i) & \\ \hline 110111 & \text{součet } (s_i) & = \text{ hodnota } -9 \text{ v doplňkovém kódu} \end{array}$$

- Protože hodnota $v = c_6 \oplus c_5 = 0$, součet čísel $x = 9$ a $y = -18$ lze v 6-bitovém doplňkovém kódu stanovit a k přetečení nedošlo

Odčítání v doplňkovém kódu

- Binární odečítání čísla y od čísla x v doplňkovém kódu znamená **přičítání opačné hodnoty čísla y (doplňku y) k číslu x** :

$$x - y = x + d(y)$$

Násobení mocninou 2^k v doplňkovém kódu

- **Násobení čísel** v doplňkovém kódu k -tou mocninou čísla 2 (2^k) znamená **bitový posun o k bitů vlevo**.
- Při bitovém posunu **doplníme zprava tolik bitů s hodnotou 0**, kolik jsme zleva odstranili

■ Příklad:

- V 8-bitovém doplňkovém kódu jsou uložena čísla 9 a -9. Vynásobte tato čísla hodnotami 2 (2^1), 4 (2^2) a 8 (2^3).

$$\begin{array}{lclclcl} 9 & \rightarrow & 00001001 & \cdot 2 & = & 00010010 & = 18 \\ & & & \cdot 4 & = & 00100100 & = 36 \\ & & & \cdot 8 & = & 01001000 & = 72 \\ -9 & \rightarrow & 11110111 & \cdot 2 & = & 11101110 & = -18 \\ & & & \cdot 4 & = & 11011100 & = -36 \\ & & & \cdot 8 & = & 10111000 & = -72 \end{array}$$

Dělení mocninou 2^k v doplňkovém kódu

- **Dělení čísel** v doplňkovém kódu k -tou mocninou čísla 2 (2^k) znamená **bitový posun o k bitů vpravo**.
 - Při bitovém posunu u **kladného čísla** doplníme **zleva tolik bitů s hodnotou 0**, kolik jsme zprava odstranili.
 - Při bitovém posunu u **záporného čísla** doplníme **zleva tolik bitů s hodnotou 1**, kolik jsme zprava odstranili.
 - Korektně dělit lze v n -bitovém doplňkovém kódu **maximálně mocninou 2^{n-1}** .

Násobení obecným celým kladným číslem v doplňkovém kódu

- Při **násobení čísla x číslem y** (celé kladné číslo) nejprve převedeme obě čísla do binárního kódu.
- Pro každou jedničku v binárním zápisu čísla y **vynásobíme číslo x odpovídající mocninou čísla 2**.
- Všechny **dílčí násobky** poté **sečteme**.

■ **Příklad:**

- V binárním kódu zjistíte výsledek **součinu čísel 9 a 7**.
- Číslo 7 vyjádříme jako mocniny čísla 2 ($7 = 4 + 2 + 1$):
 $9 \cdot 7 = 9 \cdot (4 + 2 + 1) = 9 \cdot 4 + 9 \cdot 2 + 9 \cdot 1 = 63$
- Provedeme součet (**postupný součet v doplňkovém kódu**) výsledných **násobků**:

$$\begin{array}{rcll} 9 \rightarrow & 00001001 & \cdot 4 & = 00100100 \\ & & \cdot 2 & = 00010010 \\ & & \cdot 1 & = 00001001 \\ & & & \hline & & & 00111111 \rightarrow 63 \end{array}$$

Dělení obecným celým kladným číslem v doplňkovém kódu

- Výsledkem **dělení celého čísla** celým nenulovým číslem **může ale nemusí být celé číslo**.
- V celočíselné aritmetice bývají k dispozici dvě operace
 - **div** – výsledkem je celá část podílu a / b – největší celé číslo, menší nebo rovno podílu
 - **mod** – zbytek po dělení – platí $(a \text{ div } b) \cdot b + (a \text{ mod } b) = a$
 - Pro výpočet div a mod existují jednoznačné algoritmy
- Chceme-li získat podíl jako racionální číslo (ne nutně celé), převedeme čísla do tvaru **s pohyblivou řádovou čárkou** a použijeme dělení v pohyblivé čárce (**FPU – Floating Point Unit**).

Čísla v pohyblivé řádové čárce

■ Princip:

$$x' = \pm m \cdot z^e$$

$$m = \sum_{i=1}^n a_i \cdot z^{-i}$$

■ z = **základ**, u dekadických čísel 10, zde 2

■ e = **exponent** ($\pm$)

■ m = **mantisa**

■ mantisa je normalizována, je-li z intervalu $\langle z^{-1}, 1 \rangle$ při $z = 2$ tedy $m \in \langle 1/2, 1 \rangle$

■ IEEE formát čísla - definice

■ 4 byty = 32 bitů

zn (1b)	e (8b)	m (23b)
---------	--------	---------

■ **zn** = **znaménko**:

■ 0 – kladné

■ 1 – záporné

■ **e** = **exponent** (kód s posunutou nulou)

■ $e \in \langle -2^7, 2^7 - 1 \rangle = \langle -128, 127 \rangle$

■ **m** = **mantisa v normalizovaném tvaru**

■ desetinné číslo v binárním tvaru, nejvyšší bit (a_1) je vždy 1 $\rightarrow$ neukládá se

■ IEEE formát čísla - vlastnosti

■ Je-li exponent $e \in \langle e_{\min}, e_{\max} \rangle$, lze zobrazit čísla

$$x \in \langle -2^{e_{\max}}, -2^{e_{\min}-1} \rangle \cup \{0\} \cup \langle 2^{e_{\min}-1}, 2^{e_{\max}} \rangle$$

přetečení

podtečení

přetečení

■ **Formát Double** - 8 bytů = 64 bitů – umožní počítat s větší přesností

■ **Zaokrouhluje se!** $\rightarrow$ Při výpočtech nelze testovat na (ostrou) rovnost nule!

■ Počítač pracuje vždy jen s racionálními čísly. Nelze v něm uložit (ani jedno jediné) reálné číslo, které není racionální přesně.

■ Neplatí přesně asociativní a distributivní zákon!

■ Příklad:

■ Určete obsah paměti po uložení čísla $x = -13,625$.

■ Hledáme takový exponent, abychom dostali mantisu v normalizovaném tvaru, tj. $m \in \langle 1/2, 1 \rangle$:

$$x = 0,8515625 \cdot 16 = 0,8515625 \cdot 2^4 \rightarrow \text{je splněno } m \in \langle 1/2, 1 \rangle$$

■ Nyní již lze podle předpisu zjistit hodnoty jednotlivých bitů čísla $x = -13,625$

■ Znaménko $z = 1$ (záporné číslo)

■ Exponent $e = 4 \rightarrow 4 = e' - 2^7 \rightarrow e' = 4 + 128 \rightarrow e' = 132$, binárně 10000100

■ Mantisa $m = 0,8515625 \rightarrow m = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 0 \cdot 2^{-6} + 1 \cdot 2^{-7}$,
– binárně 1101101000...0 (1.bit se neukládá)

- Obsah paměti $x = -13,625$ lze vyjádřit jako posloupnost bitů **110000100101101000...0**. V šestnáctkové soustavě vyjádřeno **(C2 5A 00 00)₁₆**.

Interpretace číselných dat

- Číselná data reprezentují informace o světě prostřednictvím **měření** (popis atributů (vlastností) objektů reálného světa čísly)
- Interpretace závisí na typu **měřicí stupnice**. Nejčastější jsou:
 - **absolutní** – číslo popisuje vlastnost jednoznačně, např. **procenta**
 - **poměrová** – poměr k etalonní hodnotě, např. **délka v metrech, kilometrech, mílích**
 - **intervalová** – proměnný počátek i etalon, např. **teplota v °C, °F, K**
 - **ordinální** – důležité je jen pořadí, např. **klasifikace ve škole**
 - **nominální** – číslo jen označuje kategorii, např. **linky veřejné dopravy**