Organizace předmětu, číselné soustavy, pevná řádová čárka a kódování čísel

ISU-cv01

Ing. Jakub Husa

Vysoké Učení Technické v Brně, Fakulta informačních technologií
Božetěchova 1/2. 612 66 Brno - Královo Pole

ihusa@fit.vutbr.cz

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

13. února 2024

Organizace předmětu

Proč se učíme ISU?

Aby mohl být program vykonán na procesoru musí být nejprve přeložen z programovacího jazyka, kterému rozumí programátor, do jazyka symbolických adres (instrukcí), lidově zvaného assembler, kterému rozumí procesor:

- V předmětu ISU se budeme učit psát programy přímo v jazyce procesoru, abychom poznali jak procesor funguje a porozuměli jeho fyzickým omezením.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}
```

gcc -S main.c

```
main:
.LFB2:
    .cfi_startproc
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    movl    $.LC0, %edi
    call    puts
    movl    $0, %eax
    popq    %rbp
    .cfi_def_cfa 7, 8
    ret
```

Organizace výuky

Garantem předmětu je doktor Filip Orság (orsag@fit.vut.cz):

- Dalšími přednášejícími jsou doktorka Marta Jaroš, a inženýři Jakub Husa a Tomáš Golmann.

Výuku cvičení zajišťuje 15 různých cvičících:

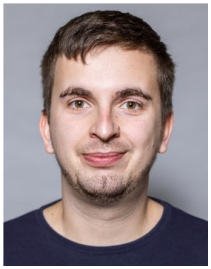
- Vaším cvičícím jsem já (Jakub Husa), najdete mě v kanceláři L307, konzultace poskytuji po dohodě e-mailem (ihusa@fit.vut.cz).

Hlavním zdrojem informací pro vás bude E-learning:

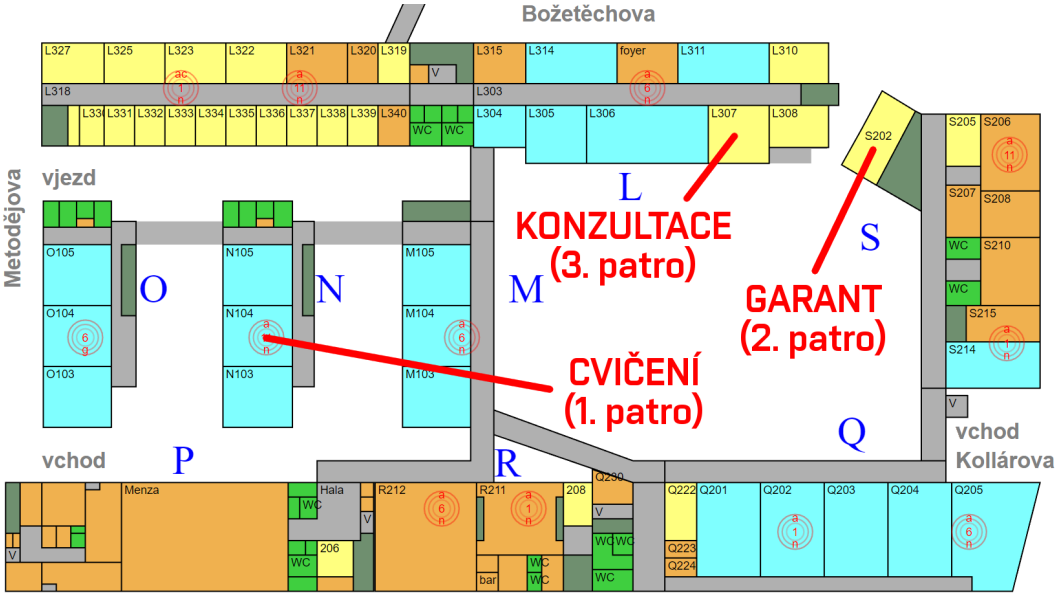
- Najdete zde obsah přednášek a cvičení, návody, vývojové nástroje, a spoustu dalších materiálů.
- Sajdy ze cvičení najdete mezi materiály cvičících.
- Jako referenční stroj budeme používat ISU-HUB.
- Aktuality mohou být zveřejňovány také ve Studisu.

Pro oficiální komunikaci vždy používejte váš školní e-mail:

- Pro neoficiální komunikaci můžete používat Discord.



Kanceláře a učebny



Hodnocení předmětu

Během semestru budete psát tři bodované testy a půlsemestrální zkoušku:

- 6. týden (místo cvičení) – 1. test (6b).
- 7. týden (místo přednášky) – půlsemestrálka (16b).
- 10. týden (místo cvičení) – 2. test (9b).
- 13. týden (místo cvičení) – 3. test (9b).
- Pokud se vám během semestru podaří získat zápočet (20b ze 40), budete na konci psát také semestrální zkoušku (min 25b ze 60).

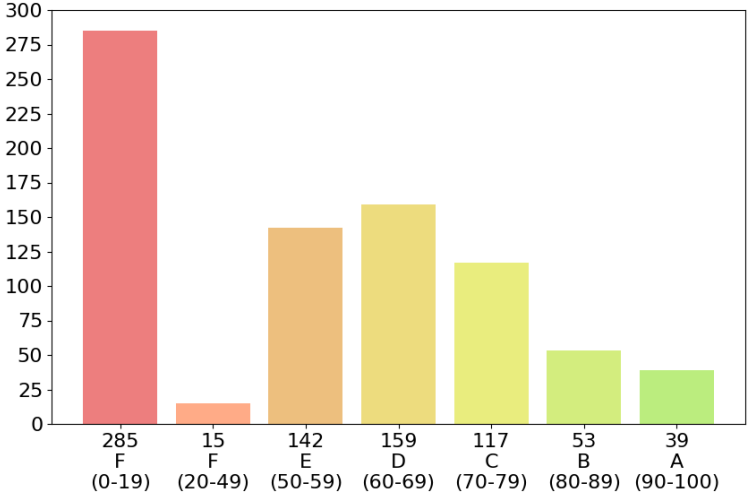
Během semestru můžete za aktivitu na cvičeních získat až 5 bonusových bodů:

- Bonusové body vám mohou zlepšit známku, ale nepočítají se do zápočtu, nemohou vaši známku změnit z F na E, a nemůžete získat více jak 1 bod za týden.

Vaše přítomnost ve výuce je **ZCELA DOBROVOLNÁ**:

- Pokud cvičení zmeškáte, můžete si ho nahradit na jiném z mých termínů, (Út–Čt, 14:00–20:00 v N104), nebo po dohodě e-mailem i u jiného cvičícího.
- Na bodované testy ale musíte přijít na zapsaný termín!

Výsledky studentů z minulého roku



Minulý rok ISU nezvládlo 37% studentů,
95% z nich proto že nezískali zápočet!

Úmrtnost studentů
v 1. semestru (2023):

- IDM – 40.7%
- IEL – 26.8%
- ILG – 29.4%
- IUS – 42.0%
- IZP – 20.8%

Úmrtnost studentů
v 2. semestru (2023):

- IMA1 – 49.8%
- INC – 32.4%
- IOS – 47.0%
- ISU – 37.0%
- IZLO – 31.8%

Číselné soustavy

Čísla obvykle reprezentujeme posloupností číslic v poziční číselné soustavě s nějakým základem:

- Nejčastěji používáme desítkovou soustavu se základem 10 a číslicemi {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- Na strojové úrovni se obvykle používá soustava dvojková, šestnáctková, a výjimečně i osmičková.

V matematice základ soustavy označujeme závorkami s pravým dolním indexem:

$$(1000001)_2 = (101)_8 = (65)_{10} = (41)_{16}$$

V programování základ soustavy označujeme předponou na začátku nebo příponou na konci čísla:

0b1000001 = 0o101 = 65 = 0x41

1000001b = 101o = 65 = 41h

Zápis čísel v různých soustavách			
2	8	10	16
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10

Převod DO desítkové soustavy

Při převodu do desítkové soustavy počítáme sumu "číslice krát základ na N-tou":

- Jednotky vždy mají mocninu na 0-tou a nalevo od nich se mocniny zvyšují.

Převod z dvojkové do desítkové:

$$(1101)_2 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 4 + 0 + 1 = (13)_{10}$$

Převod z osmičkové do desítkové:

$$(456)_8 = 4 * 8^2 + 5 * 8^1 + 6 * 8^0 = 256 + 40 + 6 = (302)_{10}$$

Převod z šestnáctkové do desítkové:

$$(123)_{16} = 1 * 16^2 + 2 * 16^1 + 3 * 16^0 = 256 + 32 + 3 = (291)_{10}$$

Vyzkoušejte si:

- Čísla převeďte do desítkové soustavy.

$$(11\,0010)_2 = (\ldots)_{10}$$

$$(555)_8 = (\ldots)_{10}$$

$$(234)_{16} = (\ldots)_{10}$$

$$(1101\,0110)_2 = (\ldots)_{10}$$

$$(1\,220)_{16} = (\ldots)_{10}$$

Převod Z desítkové soustavy

Při převodu z desítkové soustavy celočíselně dělíme základem cílové soustavy, a výsledek získáme přečtením zbytků po dělení směrem odspodu nahoru:

Převod čísel z desítkové soustavy do:

dvojkové

$(20)_{10} = (1\,0100)_2$

20	/2 =	10	zb	0
10	/2 =	5	zb	0
5	/2 =	2	zb	1
2	/2 =	1	zb	0
1	/2 =	0	zb	1

osmičkové

$(83)_{10} = (123)_8$

83	/8 =	10	zb	3
10	/8 =	1	zb	2
1	/8 =	0	zb	1

šestnáctkové

$(140)_{10} = (8C)_{16}$

140	/16 =	8	zb	12
8	/16 =	0	zb	8

pozor

$(12)_{10} = (C)_{16}$

Vyzkoušejte si:

- Čísla převeďte z desítkové soustavy.

$(180)_{10} = (\ldots)_2$
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...

$(105)_{10} = (\ldots)_2$
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...
... /2 = ... zb ...

$(93)_{10} = (\ldots)_{16}$
... /16 = ... zb ...
... /16 = ... zb ...

$(200)_{10} = (\ldots)_8$
... /8 = ... zb ...
... /8 = ... zb ...
... /8 = ... zb ...

$(357)_{10} = (\ldots)_{16}$
... /16 = ... zb ...
... /16 = ... zb ...
... /16 = ... zb ...

Přímý převod BEZ desítkové soustavy

Pokud čísla převádíme mezi dvojkovou a šestnáctkovou (nebo osmičkovou) soustavou můžeme je převést bez počítání, rozdělením na N-tice:

- Jeden znak šestnáctkové odpovídá čtyřem znakům dvojkové soustavy ($16 = 2^4$).
- Jeden znak osmičkové odpovídá třem znakům dvojkové soustavy ($8 = 2^3$)..

$$(3D0)_{16} = \begin{array}{|c|c|c|} 3 & D & 0 \\ 0011 & 1101 & 0000 \end{array} = (11\ 1101\ 0000)_2$$

- Rozdělování vždy začínáme na pozici desetinné čárky a případná prázdná místa vždy doplňujeme nulami:

$$(1\ 010\ 101\ 011)_2 = \begin{array}{|c|c|c|c|} 001 & 010 & 101 & 011 \\ 1 & 2 & 5 & 3 \end{array} = (1253)_8$$

- Nejrychlejší způsob jak číslo převést z šestnáctkové do osmičkové (a obráceně) je přes dvojkovou soustavu:

$$(7B9)_{16} = \begin{array}{|c|c|c|} 7 & B & 9 \\ 0111 & 1011 & 1001 \end{array}$$

$$\begin{array}{|c|c|c|c|} 011 & 110 & 111 & 001 \\ 3 & 6 & 7 & 1 \end{array} = (3671)_8$$

Vyzkoušejte si:

- Čísla převeďte bez desítkové soustavy.

$(1234)_8 =$ | ... | ... | ... | ... | / | ... | ... | ... | ... | $= (\ldots)_2$

$(10111000101)_2 =$ | ... | ... | ... | ... | / | ... | ... | ... | ... | $= (\ldots)_8$

$(5CE6)_{16} =$ | ... | ... | ... | ... | / | ... | ... | ... | ... | $= (\ldots)_2$

$(11110110101001)_2 =$ | ... | ... | ... | ... | / | ... | ... | ... | ... | $= (\ldots)_{16}$

$(4537)_8 =$ | ... | ... | ... | ... | / | ... | ... | ... | ... | / | ... | ... | ... | / | ... | ... | ... | $= (\ldots)_{16}$

Pevná řádová čárka

Pevná a plovoucí řádová čárka

V programování rozlišujeme mezi čísly s pevnou a plovoucí řádovou čárkou:

- Čísla s pevnou řádovou čárkou mají pevně danou přesnost a tak s nimi můžeme pracovat stejně jako s celými čísly.
- Čísla s plovoucí řádovou čárkou mají proměnlivou přesnost a mohou tak reprezentovat větší rozsah hodnot, ale práce s nimi je výrazně pomalejší.
- Přesnost obou typů desetinných čísel je omezená.

Na cvičeních ISU:

- 2. až 10. týden budeme používat pouze celá čísla.
- 11. až 13. týden budeme používat čísla s plovoucí řádovou čárkou.
- Čísla s pevnou řádovou čárkou používat nebudeme, a nemusíme se je proto učit.

Převod DO desítkové soustavy

Při převodu do desítkové soustavy počítáme sumu "číslice krát základ na N-tou":

- Jednotky vždy mají mocninu na 0-tou, nalevo od nich se mocniny zvyšují a napravo od nich se mocniny snižují.

Převod z dvojkové do desítkové:

$$(110.101)_2 = 1 * 2^2 + 1 * 2^1 + 0 * 2^0 + 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} =$$

$$= 1 * 4 + 1 * 2 + 1 * \frac{1}{2} + 1 * \frac{1}{8} =$$

$$= 4 + 2 + 0.5 + 0.125 = (6.625)_{10}$$

Převod z šestnáctkové do desítkové:

$$(AB.CD)_{16} = A * 16^1 + B * 16^0 + C * 16^{-1} + D * 16^{-2} =$$

$$= 10 * 16 + 11 * 1 + 12 * \frac{1}{16} + 13 * \frac{1}{256} =$$

$$= 160 + 11 + 0.75 + 0.05078125 = (171.80078125)_{10}$$

Převod Z desítkové soustavy

Při převodu z desítkové soustavy:

- Celou část celočíselně dělíme základem cílové soustavy a výsledek získáme přečtením zbytků po dělení směrem odspodu nahoru.
- Desetinnou část násobíme základem cílové soustavy a výsledek získáme přečtením přenosů přes desetinnou čárku směrem shora dolů.

Převod desetinného čísla z desítkové do dvojkové soustavy:

$$(12.8125)_{10} = (12)_{10} + (0.8125)_{10} = (1100)_2 + (0.1101)_2 = (1100.1101)_2$$

$(12)_{10}$ =	$(1100)_2$
12 /2 =	6 zb 0
6 /2 =	3 zb 0
3 /2 =	1 zb 1
1 /2 =	0 zb 1

$(0.8125)_{10}$ =	$(0.1101)_2$
0.8125 *2 =	1 .625
0.625 *2 =	1 .25
0.25 *2 =	0 .5
0.5 *2 =	1 .0

Převod BEZ desítkové soustavy

Pokud čísla převádíme mezi dvojkovou a šestnáctkovou (nebo osmičkovou) soustavou můžeme je převést i bez počítání, rozdělením na N-tice:

- Jeden znak šestnáctkové odpovídá čtyřem znakům dvojkové soustavy.
- Jeden znak osmičkové odpovídá třem znakům dvojkové soustavy.
- Rozdělování vždy začínáme na pozici desetinné čárky a případná prázdná místa vždy doplňujeme nulami:

$$(34.56)_8 = \left|\begin{matrix}3\\011\end{matrix}\right|\left|\begin{matrix}4\\100\end{matrix}\right|\left\|\begin{matrix}5\\101\end{matrix}\right|\left|\begin{matrix}6\\110\end{matrix}\right| = (11\ 100.101\ 11)_2$$

$$(AB.CD)_{16} = \left|\begin{matrix}A\\1010\end{matrix}\right|\left|\begin{matrix}B\\1011\end{matrix}\right|\left\|\begin{matrix}C\\1100\end{matrix}\right|\left|\begin{matrix}D\\1101\end{matrix}\right| = (1010\ 1011.1100\ 1101)_2$$

$$(11\ 100\ 110.001\ 101)_2 = \left|\begin{matrix}011\\3\end{matrix}\right|\left|\begin{matrix}100\\4\end{matrix}\right|\left|\begin{matrix}110\\6\end{matrix}\right|\left\|\begin{matrix}001\\1\end{matrix}\right|\left|\begin{matrix}101\\5\end{matrix}\right| = (346.15)_8$$

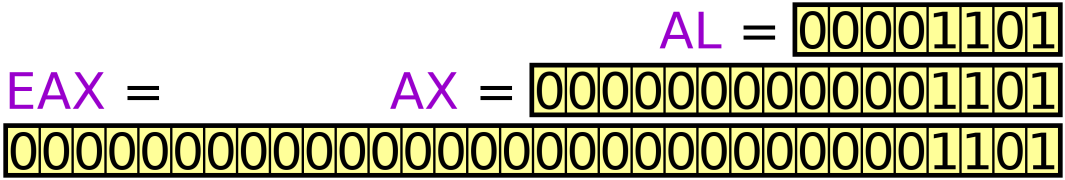
$$(10\ 0110.1011\ 111)_2 = \left|\begin{matrix}0010\\2\end{matrix}\right|\left|\begin{matrix}0110\\6\end{matrix}\right|\left\|\begin{matrix}1011\\B\end{matrix}\right|\left|\begin{matrix}1110\\E\end{matrix}\right| = (26.BE)_{16}$$

Kódování čísel

Na strojové úrovni musíme respektovat fyzická omezení architektury procesoru:

- V ISU budeme používat procesory s architekturou IA-32 (rodina x86).
- Tyto procesory celá čísla ukládají do registrů o délce 8, 16, nebo 32 bitů.
- Bity mohou obsahovat pouze nulu nebo jedničku (dvojková soustava).

Například, číslo 13 uložené v 8b registru AL, 16b registru AX, a 32b registru EAX:



Nuly na začátku registru **NELZE** vynechat:

- Protože do registrů nejde zapsat znaménko, pro reprezentaci záporných celých čísel se používá doplňkový kód.

Doplňkový kód

Doplňkový kód mění rozsah hodnot 8b registru z <0, 255> na <-128, 127>:

- Čísla začínající nulou jsou kladná, čísla začínající jedničkou jsou záporná.
- Znaménko čísla obrátíme operací dvojkového doplňku (aritmetická negace), která invertuje všechny bity a přičte jedničku.

$$
\begin{array}{rl}
(13)_{10} & = 0\,0\,0\,0\,1\,1\,0\,1 \\
\hline
\text{inv}(13) & = 1\,1\,1\,1\,0\,0\,1\,0 \\
+\,1 & = 0\,0\,0\,0\,0\,0\,0\,1 \\
\hline
 & = 1\,1\,1\,1\,0\,0\,1\,1 = (-13)_{10}
\end{array}
$$

$$
\begin{array}{rl}
(-13)_{10} & = 1\,1\,1\,1\,0\,0\,1\,1 \\
\hline
\text{inv}(-13) & = 0\,0\,0\,0\,1\,1\,0\,0 \\
+\,1 & = 0\,0\,0\,0\,0\,0\,0\,1 \\
\hline
 & = 0\,0\,0\,0\,1\,1\,0\,1 = (13)_{10}
\end{array}
$$

Vyzkoušejte si:

- Čísla v desítkové soustavě převeďte do doplňkového kódu na osmi bitech.

35 = ...

-23 = inv(23) + 1 = inv(...) + 0000 0001 = ...

-97 = inv(97) + 1 = inv(...) + 0000 0001 = ...

- Čísla v doplňkovém kódu na osmi bitech převeďte do desítkové soustavy.

0011 0110 = ...

1101 1001 = - (inv(1101 1001) + 0000 0001) = - (...) = ...

1000 1000 = - (inv(1000 1000) + 0000 0001) = - (...) = ...

Doplňkový kód umožňuje se zápornými čísly provádět aritmetické operace:

- Na cvičeních ISU si vystačíme se sčítáním a negací.

100 =	0 1 1 0 0 1 0 0	
+ 5 =	0 0 0 0 0 1 0 1	
	0 1 1 0 1 0 0 1	= 105 ok

Protože registry mají omezený rozsah, při výpočtu může dojít k přetečení (overflow):

- Při počítání bez znaménka by k přetečení došlo při přenosu **Z** nejvyššího bitu.
- V doplňkovém kódu k přetečení dojde při přenosu **DO** nebo **Z** nejvyššího bitu.

100 =	0 1 1 0 0 1 0 0	
+ 50 =	0 0 1 1 0 0 1 0	
	1 0 0 1 0 1 1 0	= -106 chyba

- Pokud v doplňkovém kódu dojde k přenosu **DO** i **Z** nejvyššího bitu, obě chyby se spolu vzájemně vyruší a výsledek bude platný!

100 =	0 1 1 0 0 1 0 0	
+ -50 =	1 1 0 0 1 1 1 0	
1	0 0 1 1 0 0 1 0	= 50 ok

Na procvičení

Vyzkoušejte si:

- Sečtěte čísla v doplňkovém kódu na osmi bitech, a rozhodněte jestli je výsledek platný nebo jestli při výpočtu došlo k přetečení.

66 =	0 1 0 0 0 0 1 0	
+ -99 =	1 0 0 1 1 1 0 1	
	. . . . .	= ...

-123 =	1 0 0 0 0 1 0 1	
+ -45 =	1 1 0 1 0 0 1 1	
	. . . . .	= ...

-70 =	1 0 1 1 1 0 1 0	
+ -58 =	1 1 0 0 0 1 1 0	
	. . . . .	= ...

70 =	0 1 0 0 0 1 1 0	
+ 58 =	0 0 1 1 1 0 1 0	
	. . . . .	= ...