

## Řídicí struktury, výrazy a funkce

Jan Faigl

Katedra počítačů  
Fakulta elektrotechnická  
České vysoké učení technické v Praze

Přednáška 03

B0B36PRP – Procedurální programování

## Přehled témat

### ■ Část 1 – Řídicí struktury

Příkaz a složený příkaz

Příkazy řízení běhu programu

Podmíněné řízení běhu programu

Konečnost cyklu

S. G. Kochan: kapitoly 5 a 6

### ■ Část 2 – Výrazy

Výrazy a operátory

Přiházení

S. G. Kochan: kapitola 4, 12

### ■ Část 3 – Zadání 3. domácího úkolu (HW03)

## Část I

## Řídicí struktury

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 1 / 60  
Příkaz a složený příkaz Příkazy řízení běhu programu Podmíněné řízení běhu programu Konečnost cyklu

## Příkaz a složený příkaz (blok)

- Příkaz je výraz zakončený středníkem. *Příkaz tvořený pouze středníkem je prázdný příkaz.*
- Blok je tvořen seznamem definic proměnných a příkazů.
- Uvnitř bloku, definice proměnných zpravidla předchází příkazům.  
*Záleží na standardu jazyka, platí pro ANSI C (C89, C90).*
- Začátek a konec bloku je vymezen složenými závorkami { a }.
- Bloky mohou být vnorené do jiného bloku.

```
1 void function(void)      1 void function(void) { /* function block start */
2 { /* function block start */ 2 { /* inner block */
3 { /* inner block */        3 for (int i = 0; i < 10; ++i) {
4 for (i = 0; i < 10; ++i)    4 //inner for-loop block
5 {                          5 }
6 //inner for-loop block    6 }
7 }                          7 }
8 }                          8 }
9 }                          9 }
```

*Různé kódovací konvence.*

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 5 / 60  
Příkaz a složený příkaz Příkazy řízení běhu programu Podmíněné řízení běhu programu Konečnost cyklu

## Složený příkaz a zanořování 1/2

**Čtyři úrovně zanoření.**

```
1 int get_sum_of_even_numbers(int from, int to)
2 {
3     if (from < to) {
4         for (int number = from; number <= to; ++number) {
5             if (number % 2 == 0) {
6                 sum += number;
7             }
8         } // end for loop
9         return sum;
10    } else {
11        return 0;
12    }
13 }
14
```

*Mírně na čitelnější podobu.*

```
1 int get_sum_of_even_numbers(int from, int to)
2 {
3     if (from > to) return 0;
4     int sum = 0;
5     for (int number = from; number <= to; ++number) {
6         sum += filter_odd(number);
7     } // end for loop
8     return sum;
9 }
```

**Vyjmutí (definice nové funkce).**

```
1 int filter_odd(int number);
2 int get_sum_of_even_numbers(int from, int to)
3 {
4     if (from < to) {
5         int sum = 0;
6         for (int number = from; number <= to; ++number) {
7             sum += filter_odd(number);
8         } // end for loop
9         return sum;
10    } else {
11        return 0;
12    }
13 }
14
```

**Finální „zkompatnění“.**

```
1 int filter_odd(int number);
2 int get_sum_of_even_numbers(int from, int to)
3 {
4     if (from > to) return 0;
5     int sum = 0;
6     for (int number = from; number <= to; ++number) {
7         sum += filter_odd(number);
8     } // end for loop
9     return sum;
10 }
11
```

**Inverze (záměna podminky hodnoty vstupu).**

```
1 int filter_odd(int number);
2 int get_sum_of_even_numbers(int from, int to)
3 {
4     if (from > to) {
5         return 0;
6     }
7     int sum = 0;
8     for (int number = from; number <= to; ++number) {
9         sum += filter_odd(number);
10    } // end for loop
11    return sum;
12 }
13
```

**Finální „zkompatnění“.**

```
1 int filter_odd(int number);
2 int get_sum_of_even_numbers(int from, int to)
3 {
4     if (from > to) return 0;
5     int sum = 0;
6     for (int number = from; number <= to; ++number) {
7         sum += filter_odd(number);
8     } // end for loop
9     return sum;
10 }
11
```

**Použitím technik vyjmutí a inverze redukuje počet zanoření.**

```
1 int filter_odd(int number)
2 {
3     if (number % 2 == 0) {
4         return number;
5     }
6     return 0;
7 }
8
```

**Použitím technik vyjmutí a inverze redukuje počet zanoření.**

```
1 int filter_odd(int number)
2 {
3     if (number % 2 == 0) {
4         return number;
5     }
6     return 0;
7 }
8
```

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 8 / 60

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 2 / 60  
Příkaz a složený příkaz Příkazy řízení běhu programu Podmíněné řízení běhu programu Konečnost cyklu

## Srozumitelnost, čitelnost kódu - kódovací konvence a styl (čistota kódu)

- Konvence a styl je důležitý, protože podporuje přehlednost a čitelnost.  
[https://www.gnu.org/prep/standards/html\\_node/Writing-C.html](https://www.gnu.org/prep/standards/html_node/Writing-C.html)
  - Formátování patří k úplným základům. *Nastavte si automatické formátování v textovém editoru.*
  - Volba výstižného jména identifikátorů podporuje čitelnost.  
*Co může být jasně nyní, za pár dní či měsíců může být jinak.*
  - Cvičte se v kódovací konvenci a zvoleném stylu i za cenu zdánlivě pomalejšího zápisu kódu. *Přehlednost je důležitá, zvláště pokud hledáte chybu.*
  - Doporučená konvence v rámci PRP  
*Nezřídka je užitečné nebýt se začít úplně znovu a lépe.*
    - Píšte zdrojové kódy pokud možno anglicky (identifikátory).
    - Pro proměnné volte podstatná jména.
    - Pro funkce volte slovesa.
- Osobní preference přednášejícího: odsazení 3 znaky, mezery místo tabulátorů.*

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 6 / 60  
Příkaz a složený příkaz Příkazy řízení běhu programu Podmíněné řízení běhu programu Konečnost cyklu

## Složený příkaz a zanořování 2/2

**Inverze (záměna podminky hodnoty vstupu).**

```
1 int filter_odd(int number);
2 int get_sum_of_even_numbers(int from, int to)
3 {
4     if (from > to) {
5         return 0;
6     }
7     int sum = 0;
8     for (int number = from; number <= to; ++number) {
9         sum += filter_odd(number);
10    } // end for loop
11    return sum;
12 }
13
```

**Finální „zkompatnění“.**

```
1 int filter_odd(int number);
2 int get_sum_of_even_numbers(int from, int to)
3 {
4     if (from > to) return 0;
5     int sum = 0;
6     for (int number = from; number <= to; ++number) {
7         sum += filter_odd(number);
8     } // end for loop
9     return sum;
10 }
11
```

**Použitím technik vyjmutí a inverze redukuje počet zanoření.**

```
1 int filter_odd(int number)
2 {
3     if (number % 2 == 0) {
4         return number;
5     }
6     return 0;
7 }
8
```

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 9 / 60

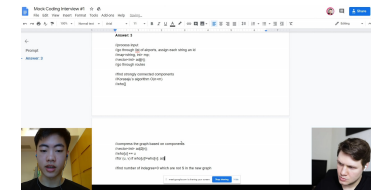
Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 3 / 60  
Příkaz a složený příkaz Příkazy řízení běhu programu Podmíněné řízení běhu programu Konečnost cyklu

## Srozumitelnost a čitelnost kódu - kódovací konvence

- Existují různé kódovací konvence; inspiруйте se existujícími doporučeními a čtením reprezentativních kódů.



Clean Code - Uncle Bob / Lesson 1  
<https://youtu.be/7EmboKQH81M>



Google Coding Interview with a High School Student  
<https://youtu.be/qz9tKlF431k>

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 7 / 60  
Příkaz a složený příkaz Příkazy řízení běhu programu Podmíněné řízení běhu programu Konečnost cyklu

## Příkazy řízení běhu programu

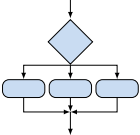
- Podmíněné řízení běhu programu
  - Podmíněný příkaz: `if ()` nebo `if () ... else`
  - Programový přepínač: `switch () case ...`
- Cykly
  - `for ()`
  - `while ()`
  - `do ... while ()`
- Nepodmíněné větvení programu
  - `continue`
  - `break`
  - `return`
  - `goto`

Jan Faigl, 2025 B0B36PRP – Přednáška 03: Řídicí struktury, výrazy a funkce 11 / 60

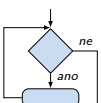
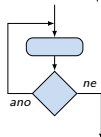
| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Podmíněné větvení – if                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ <b>if</b> (výraz) <b>příkaz1</b>; <b>else</b> <b>příkaz2</b></li><li>■ Je-li hodnota výrazu <b>výraz</b> <b>!= 0</b> (TRUE), provede se příkaz <b>příkaz1</b> jinak <b>příkaz2</b>.</li><li>■ Část <b>else</b> je nepovinná.</li><li>■ Podmíněné příkazy mohou být vnořené a můžeme je řetězit.</li></ul> <p><i>Příkaz může být blok příkazů.</i></p> <pre>1 int max; 2 if (a &gt; b) { 3     ... 4 } else if (a &lt; c) { 5     ... 6 } else if (a == b) { 7     ... 8 } else { 9     ... 10 }</pre> <ul style="list-style-type: none"><li>■ Příklad zápisu</li></ul> <pre>1 if (x &lt; y) { 2     int tmp = x; 3     x = y; 4     y = tmp; 5 }</pre> <p><i>Jaký je smysl těchto programů?</i></p> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 13 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                       | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Programový přepínač switch – Příklad                                                                                                                                                                                                                                                                                                                                                                                                          |                                                            |                                |                 |
| <pre>1 switch (v) { 2     case 'A': 3         printf("Upper 'A'\n"); 4         break; 5     case 'a': 6         printf("Lower 'a'\n"); 7         break; 8     default: 9         printf("It is not 'A' nor 'a'\n"); 10        break; 11 }</pre> <pre>1 if (v == 'A') { 2     printf("Upper 'A'\n"); 3 } else if (v == 'a') { 4     printf("Lower 'a'\n"); 5 } else { 6     printf("It is not 'A' nor 'a'\n"); 7 }</pre> <p>lec03/switch.c</p> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                               | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 16 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Podmíněný výraz / ternární operátor                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ V příkladu hledání největšího společného dělitele můžeme podmíněné nastavení proměnné <b>d</b> větvením <b>if</b> a <b>else</b> nahradit podmíněným výrazem <b>?</b> :</li></ul> <pre>1 int getGreatestCommonDivisor(int x, int y) 2 { 3     int d; 4     if (x &lt; y) { 5         d = x; 6     } else { 7         d = y; 8     } 9     while ( (x % d != 0)    (y % d != 0) ) { 10        d = d - 1; 11    } 12    return d; 13 }</pre> <ul style="list-style-type: none"><li>■ Této lze zkráceně dosáhnout podmíněným výrazem <b>expr<sub>1</sub> ? expr<sub>2</sub> : expr<sub>3</sub></b>.</li></ul> <pre>1 int getGreatestCommonDivisor(int x, int y) 2 { 3     int d = x &lt; y ? x : y; 4     while ( (x % d != 0)    (y % d != 0) ) { 5         d = d - 1; 6     } 7     return d; 8 }</pre> <p>lec03/demo-gcd.c</p> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 19 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Příkaz větvení switch                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ Příkaz <b>switch</b> (přepínač) umožňuje větvení programu do více větví na základě různých hodnot výrazu výtčového (celočíselného) typu, jako jsou např. <b>int</b>, <b>char</b>, <b>short</b>, <b>enum</b>.</li><li>■ Základní tvar příkazu.</li></ul> <pre>switch (výraz) {     case konstanta<sub>1</sub>: příkazy<sub>1</sub>; break;     case konstanta<sub>2</sub>: příkazy<sub>2</sub>; break;     ...     case konstanta<sub>n</sub>: příkazy<sub>n</sub>; break;     default: příkazy<sub>def</sub>; break; }</pre> <p>kde <i>konstanty</i> jsou téhož typu jako <i>výraz</i> a <i>příkazy</i>; jsou posloupnosti příkazů.</p> <ul style="list-style-type: none"><li>■ Příkaz větvení <b>switch</b> mohou být vnořené.</li></ul> <p><i>Sémantika: vypočte se hodnota výrazu a provedou se ty příkazy, které jsou označeny konstantou s identickou hodnotou. Není-li vybrána žádná větev, provedou se příkazy<sub>def</sub> (pokud jsou uvedeny).</i></p>  |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 14 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Větvení switch – pokračování ve vykonávání dalších větví                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ Příkaz <b>break</b> dynamicky ukončuje větev, pokud jej neuvedeme, pokračuje se v provádění další větve.</li></ul> <pre>1 int part = ? 2 switch(part) { 3     case 1: 4         printf("Branch 1\n"); 5         break; 6     case 2: 7         printf("Branch 2\n"); 8     case 3: 9         printf("Branch 3\n"); 10        break; 11    case 4: 12        printf("Branch 4\n"); 13        break; 14    default: 15        printf("Default branch\n"); 16        break; 17 }</pre> <ul style="list-style-type: none"><li>■ part ← 1<br/>Branch 1</li><li>■ part ← 2<br/>Branch 2<br/>Branch 3</li><li>■ part ← 3<br/>Branch 3</li><li>■ part ← 4<br/>Branch 4</li><li>■ part ← 5<br/>Default branch</li></ul> <p>lec03/demo-switch_break.c</p> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 17 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Cykly                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ Cyklus <b>for</b> a <b>while</b> testuje podmínku opakování před vstupem do těla cyklu.</li></ul> <ul style="list-style-type: none"><li>■ <b>for</b> – inicializace, podmínka a změna řídicí proměnné.</li></ul> <pre>1 for (int i = 0; i &lt; 5; ++i) { 2     ... 3 }</pre> <ul style="list-style-type: none"><li>■ <b>while</b> – řídicí proměnná v těle cyklu.</li></ul> <pre>1 int i = 0; 2 while (i &lt; 5) { 3     ... 4     i += 1; 5 }</pre> <ul style="list-style-type: none"><li>■ Cyklus <b>do</b> testuje podmínku opakování cyklu po prvním provedení cyklu.</li></ul> <pre>1 int i = -1; 2 do { 3     ... 4     i += 1; 5 } while (i &lt; 5);</pre>   |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 20 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Programový přepínač – switch                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ Přepínač <b>switch(výraz)</b> větví program do <i>n</i> směrů.</li><li>■ Hodnota <b>výraz</b> je porovnávána s <i>n</i> konstantními výrazy typu <b>int</b> příkazy.</li></ul> <pre>case konstanta<sub>1</sub>: ...</pre> <ul style="list-style-type: none"><li>■ Hodnota <b>výraz</b> musí být celočíselná a hodnoty <b>konstanta<sub>1</sub></b> musejí být navzájem různé.</li><li>■ Pokud je nalezena shoda, program pokračuje od tohoto místa dokud nenajde příkaz <b>break</b> nebo konec příkazu <b>switch</b>.</li><li>■ Pokud shoda není nalezena, program pokračuje nepovinnou sekcí <b>default</b>.</li></ul> <p><i>Seke default se zpravidla uvádí jako poslední.</i></p> <ul style="list-style-type: none"><li>■ Příkazy <b>switch</b> mohou být vnořené.</li></ul> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 15 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Příklad větvení switch vs if-then-else                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ Napište konverzní program, který podle čísla dnu v týdnu vytiskne na obrazovku jméno dne. Ošetřete případ, kdy bude zadané číslo mimo platný rozsah (1 až 7).</li></ul> <pre>1 int day_of_week = 3; 2 if (day_of_week == 1) { 3     printf("Monday"); 4 } else if (day_of_week == 2) { 5     printf("Tuesday"); 6 } else ... { 7     printf("Tuesday"); 8 } else if (day_of_week == 7) { 9     printf("Sunday"); 10 } else { 11     fprintf(stderr, "Invalid number"); 12 }</pre> <pre>1 int day_of_week = 3; 2 switch (day_of_week) { 3     case 1: 4         printf("Monday"); 5         break; 6     case 2: 7         printf("Tuesday"); 8         break; 9     ... 10    case 7: 11        printf("Sunday"); 12        break; 13    default: 14        fprintf(stderr, "Invalid number"); 15        break; 16 }</pre> <p>lec03/demo-switch_day_of_week.c</p> <p><i>Oba způsoby jsou sice funkční, nicméně elegantnější lze vyřešit úlohu použitím datové struktury pole nebo ještě lépe asociativním polem / hash mapou.</i></p> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 18 / 60         |

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Příkazy řízení běhu programu                               | Podmíněné řízení běhu programu | Konečnost cyklu |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|--------------------------------|-----------------|
| Cyklus for                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                            |                                |                 |
| <ul style="list-style-type: none"><li>■ Základní příkaz cyklu <b>for</b> má tvar <b>for</b> (<i>inicializace; podmínka; změna</i>) <i>příkaz</i>.</li><li>■ Odpovídá cyklu <b>while</b> ve tvaru:</li></ul> <pre>inicializace; while (podmínka) {     příkaz;     změna; }</pre> <ul style="list-style-type: none"><li>■ Změnu řídicí proměnné lze zkráceně zapsat operátorem inkrementace nebo dekrementace <b>++</b> a <b>--</b>.</li><li>■ Alternativně lze též použít zkrácený zápis přiřazení, např. <b>+=</b>.</li></ul> <pre>1 for (int i = 0; i &lt; 10; ++i) { 2     printf("i: %i\n", i); 3 }</pre> |                                                            |                                |                 |
| Jan Faigl, 2025                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | BOB36PRP – Prednáška 03: Řídicí struktury, výrazy a funkce |                                | 21 / 60         |

Cyklus while a do-while

- Základní příkaz cyklu **while** má tvar **while** (*podmínka*) příkaz;
- Základní příkaz cyklu **do-while** má tvar **do** příkaz **while** (*podmínka*).

Příklad

```
1 q = x;
2 while (q >= y) {
3     q = q - y;
4 }
```

```
1 q = x;
2 do {
3     q = q - y;
4 } while (q >= y);
```

- Jaká je hodnota proměnné *q* po skončení cyklu pro hodnoty.
  - $x \leftarrow 10$  a  $y \leftarrow 3$
  - $x \leftarrow 2$  a  $y \leftarrow 3$

```
while: 1, do-while: 1
while: 2, do-while: -1
lec03/demo-while.c
```

Příkaz break – Ukončení cyklu

- Příkaz **break** způsobí opuštění těla cyklu. Program pokračuje dalším příkazem za tělem cyklu.

- Příklad **break** ve **while** cyklu.

```
1 int i = 10;
2 while (i > 0) {
3     if (i == 5) {
4         printf("i reaches 5, leave the loop\n");
5         break;
6     }
7     i--;
8     printf("End of the while loop i: %d\n", i);
9 }
```

```
1 $ clang break.c
2 $ ./a.out
3 End of the while loop i: 9
4 End of the while loop i: 8
5 End of the while loop i: 7
6 End of the while loop i: 6
7 End of the while loop i: 5
8 i reaches 5, leave the loop
9
lec03/break.c
```

- Příkazy **continue** a **break** ve **for** cyklu.

```
1 for (int i = 0; i < 10; ++i) {
2     printf("i: %i ", i);
3     if (i % 3 != 0) {
4         continue;
5     }
6     printf("\n");
7     if (i > 5) {
8         break;
9     }
10 }
```

```
$ clang demo-break
.c
$ ./a.out
i:1 i:2 i:3
i:4 i:5 i:6
lec03/demo-break.c
```

Příklad goto

```
1 #include <stdio.h>
2 int main(void)
3 {
4     for (int i = 0; i < 5; ++i) {
5         for (int j = 0; j < 3; ++j) {
6             printf("i-j: %i-%i\n", i, j);
7             if (j == 2) {
8                 goto outer;
9             }
10        }
11    }
12 }
13 outer:
14 int a;
15 return 0;
16 }
```

```
1 #include <stdio.h>
2 int main(void)
3 {
4     for (int i = 0; i < 5; ++i) {
5         for (int j = 0; j < 3; ++j) {
6             printf("i-j: %i-%i\n", i, j);
7             if (j == 2) {
8                 goto outer;
9             }
10        }
11    }
12 }
13 outer;; // za návěštím musí být příkaz
14 int a; // definice není příkaz
15 return 0;
16 }
```

lec03/goto\_outer.c

Cyklus for( ; ; )

- Příkaz **for** cyklu má tvar **for** ([*vyraz<sub>1</sub>*]; [*vyraz<sub>2</sub>*]; [*vyraz<sub>3</sub>*]) příkaz;
- Cyklus **for** používá řídicí proměnnou a probíhá následovně:
  1. *vyraz<sub>1</sub>* – Inicializace (zpravidla řídicí proměnné);
  2. *vyraz<sub>2</sub>* – Test řídicího výrazu;
  3. Pokud *vyraz<sub>2</sub>* !=0 provede se příkaz, jinak cyklus končí;
  4. *vyraz<sub>3</sub>* – Aktualizace proměnných na konci běhu cyklu;
  5. Opakování cyklu testem řídicího výrazu.
- Výrazy *vyraz<sub>1</sub>* a *vyraz<sub>3</sub>* mohou být libovolného typu.
- Libovolný z výrazů lze vynechat.
- **break** – cyklus lze nuceně opustit příkazem **break**.
- **continue** – část těla cyklu lze vynechat příkazem **continue**.

Příkaz přeruší vykonávání těla (blokového příkazu) pokračuje vyhodnocením *vyraz<sub>3</sub>*.

- Při vynechání řídicího výrazu *vyraz<sub>2</sub>* se cyklus bude provádět nepodmíněně.  
`for (;;) {...}`

Nekonečný cyklus

Příkaz goto

- Příkaz nepodmíněného lokálního skoku **goto** předá řízení na místo určené návěštím *navesti* – syntax **goto** *navesti*;
- Návěští má tvar *navesti* příkaz.
- Příkaz **goto** lze použít pouze v těle funkce a skok je možný pouze rámci jediné funkce.

Definice proměnné není příkaz.

```
1 int test = 3;
2 for (int i = 0; i < 3; ++i) {
3     for (int j = 0; j < 5; ++j) {
4         if (j == test) {
5             goto loop_out;
6         }
7         fprintf(stdout, "Loop i: %d j: %d\n", i, j);
8     }
9 }
10 return 0;
11 loop_out:
12 fprintf(stdout, "After loop\n"); // goto can jump to a label that
13     represents statement (there must be an address to be jumped at).
14 return -1;
```

lec03/goto.c

Konečnost cyklů 1/3

- Konečnost algoritmu – pro přípustná data v konečné době skončí.
- Aby byl algoritmus **konečný** musí každý cyklus v něm uvedený skončit po konečném počtu kroků.
- Jedním z důvodů neukončení programu je zacyklení.
  - Program opakovaně vykoná cyklus, jehož podmínka ukončení není nikdy splněna.

```
1 while (i != 0) {
2     j = i - 1;
3 }
```

```
1 while (i != 0) {
2     j = i - 1;
3 }
```

Příkaz continue

- Příkaz návratu na vyhodnocení řídicího výrazu – **continue**.
- Příkaz **continue** lze použít pouze v těle cyklů.
  - **for** ()
  - **while** ()
  - **do...while** ()

```
1 int i;
2 for (i = 0; i < 20; ++i) {
3     if (i % 2 == 0) {
4         continue;
5     }
6     printf("%3d", i);
7 }
```

```
1 for (int i = 0; i < 10; ++i) {
2     printf("i: %i ", i);
3     if (i % 3 != 0) {
4         continue;
5     }
6     printf("\n");
7 }
```

```
1 clang demo-continue.c
2 ./a.out
3 i:0
4 i:1 i:2 i:3
5 i:4 i:5 i:6
```

Vnořené cykly

- Příkaz **break** ukončuje vnitřní cyklus.

```
1 for (int i = 0; i < 3; ++i) {
2     for (int j = 0; j < 3; ++j) {
3         printf("i-j: %i-%i\n", i, j);
4         if (j == 1) {
5             break;
6         }
7     }
8 }
```

```
i-j: 0-0
i-j: 0-1
i-j: 1-0
i-j: 1-1
i-j: 2-0
i-j: 2-1
```

- Vnější cyklus můžeme ukončit příkazem **goto**.

```
1 for (int i = 0; i < 5; ++i) {
2     for (int j = 0; j < 3; ++j) {
3         printf("i-j: %i-%i\n", i, j);
4         if (j == 2) {
5             goto outer;
6         }
7     }
8 }
9 outer:
```

```
i-j: 0-0
i-j: 0-1
i-j: 0-2
```

lec03/demo-goto.c

Konečnost cyklů 2/3

- Základní pravidlo pro konečnost cyklu.
  - Provedení těla cyklu se musí změnit hodnota proměnné použité v podmínce ukončení cyklu.

```
1 for (int i = 0; i < 5; ++i) {
2     ...
3 }
```

- Uvedené pravidlo konečnost cyklu nezaručuje.

```
1 int i = -1;
2 while ( i < 0 ) {
3     i = i - 1;
4 }
```

Konečnost cyklu závisí na hodnotě proměnné před vstupem do cyklu.

| Příkaz a složený příkaz                                                                                                                                                                                     | Příkazy řízení běhu programu | Podmíněné řízení běhu programu    | Konečnost cyklu                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Konečnost cyklů 3/3                                                                                                                                                                                         |                              |                                   |                                                                                                                                                                                                                                                                                                                                                     |
| <pre>1 while (i != n) { 2   ... //příkazy nemenící hodnotu promenne i 3   i++; 4 }</pre>                                                                                                                    |                              | <pre>lec03/demo-loop_byte.c</pre> | <ul style="list-style-type: none"><li>Vstupní podmínka konečnosti uvedeného cyklu.<ul style="list-style-type: none"><li><math>i \leq n</math> pro celá čísla.</li></ul></li></ul> <p><i>Jak by vypadala podmínka pro proměnné typu double?</i><br/><i>Co se stane pokud by proměnná i byla typu unsigned char?</i></p> <pre>lec03/demo-loop.c</pre> |
| <ul style="list-style-type: none"><li>Splnění vstupní podmínky konečnosti cyklu musí zajistit příkazy předcházející příkazu cyklu.</li><li>Zabezpečený program testuje přípustnost vstupních dat.</li></ul> |                              |                                   |                                                                                                                                                                                                                                                                                                                                                     |

### Příklad – test zda-li je číslo prvočíslem – isPrimeNumber() 3/3

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <ul style="list-style-type: none"><li>Misto příkazu <b>break</b> můžeme testovat hodnotu <b>ret</b> v podmínce opakování cyklu.</li></ul> <pre>1 _Bool isPrimeNumber(int n) 2 { 3   _Bool ret = true; 4   const int maxBound = (int)sqrt((double)n); 5   for (int i = 2; ret &amp;&amp; i &lt;= maxBound; ++i) { 6     if (n % i == 0) { 7       ret = false; // leave the loop if it is sure n is not a prime number 8     } 9   } 10  return ret; 11 }</pre> <ul style="list-style-type: none"><li>Nebo ještě přidat nastavení <b>ret</b> při aktualizaci hodnoty řídící proměnné.</li></ul> <pre>1 _Bool isPrimeNumber(int n) 2 { 3   _Bool ret = true; 4   const int maxBound = (int)sqrt((double)n); 5   for (int i = 2; ret &amp;&amp; i &lt;= maxBound; ret = !(n % i++ == 0)); 6   return ret; 7 }</pre> | <p>Čitelnost je v tomto případě diskutabilní.</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|

### Výrazy

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none"><li><b>Výraz</b> předepisuje výpočet hodnoty určitého vstupu.</li><li>Struktura výrazu obsahuje <i>operandy</i>, <i>operátory</i> a <i>závorky</i>.</li><li>Výraz může obsahovat:<ul style="list-style-type: none"><li>literály;</li><li>proměnné;</li><li>konstanty;</li><li>unární a binární operátory;</li><li>volání funkcí;</li><li>závorky.</li></ul></li><li>Pořadí operací předepsaných výrazem je dáno <b>prioritou</b> a <b>asociativitou</b> operátorů.</li></ul> <p><b>Příklad</b></p> <pre>10 + x * y // pořadí vyhodnocení 10 + (x * y) 10 + x + y // pořadí vyhodnocení (10 + x) + y</pre> <p><i>* má vyšší prioritu než + je asociativní zleva</i></p> |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

| Příkaz a složený příkaz                                                                                                                                                                                                                                                                                                                       | Příkazy řízení běhu programu | Podmíněné řízení běhu programu | Konečnost cyklu                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Příklad – test zda-li je číslo prvočíslem – isPrimeNumber() 1/3                                                                                                                                                                                                                                                                               |                              |                                |                                                                                                                     |
| <pre>1 #include &lt;stdbool.h&gt; 2 #include &lt;math.h&gt; 4 _Bool isPrimeNumber(int n) 5 { 6   _Bool ret = true; 7   for (int i = 2; i &lt;= (int)sqrt((double)n); ++i) { 8     if (n % i == 0) { 9       ret = false; // leave the loop once it is sure 10      break; // n is not a prime number 11    } 12  } 13  return ret; 14 }</pre> |                              | <pre>lec03/demo-prime.c</pre>  | <ul style="list-style-type: none"><li><b>break</b> – po nalezení prvního dělitele nemusíme dále testovat.</li></ul> |

### Kódovací konvence

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none"><li>Příkazy <b>break</b> a <b>continue</b> v podstatě odpovídají příkazům skoku.</li><li>Obecně můžeme říci, že příkazy <b>break</b> a <b>continue</b> nepřidávají příliš na přehlednosti.<p><i>Ne nutné break v příkazu switch.</i></p></li><li>Prerušení cyklu <b>break</b> nebo <b>continue</b> můžeme využít v těle dlouhých funkcí a vnořených cyklech.<p><i>Ale funkce bychom měli psát krátké a přehledné.</i></p></li><li>Je-li funkce (tělo cyklu) krátké, je význam <b>break/continue</b> čitelný.</li><li>Podobné použití na začátku bloku cyklu, např. jako součást testování splnění předpokladů, je zpravidla přehledné.</li><li>Použití uprostřed bloku je však už méně přehledné a může snížit čitelnost a porozumění kódu.</li></ul> <p><a href="https://www.scribd.com/doc/38873257/Knuth-1974-Structured-Programming-With-Go-to-Statements">https://www.scribd.com/doc/38873257/Knuth-1974-Structured-Programming-With-Go-to-Statements</a></p> |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

### Výrazy a operátory

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none"><li>Výraz se skládá z operátorů a operandů.<ul style="list-style-type: none"><li>Nejjednodušší výraz tvoří konstanta, proměnná nebo volání funkce.</li><li>Výraz sám může být operandem.</li><li>Výraz má <b>typ</b> a <b>hodnotu</b>.<p><i>(Pouze výraz typu void hodnotu nemá.)</i></p></li><li>Výraz zakončený středníkem ; je příkaz.</li></ul></li><li>Operátory jsou vyhrazené znaky pro zápis výrazů.<p><i>Případné posloupnost znaků.</i></p></li><li>Postup výpočtu výrazu s více operátory je dán prioritou operátorů.<p><i>Postup výpočtu lze předepsat použitím kulatých závorek ( a ).</i></p></li><li>Operátory: aritmetické, relační, logické, bitové.<ul style="list-style-type: none"><li>Arita operátoru (počet operandů) – unární, binární, ternární.</li><li>Obecně (mimo konkrétní případy) není pořadí vyhodnocení operandů definováno (nezaměňovat s asociativitou).<p><i>Např. pro součet f1() + f2() není definováno, který operand se vyhodnotí jako první (jaká funkce se zavolá jako první).</i></p></li><li>Chování i = ++i + i++; není definováno, závisí na překladači.</li></ul></li><li>Pořadí vyhodnocení je <b>definováno pro operandy v logickém součinu AND a součtu OR</b>.<p><a href="http://en.cppreference.com/w/c/language/eval_order">http://en.cppreference.com/w/c/language/eval_order</a></p></li></ul> |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

| Příkaz a složený příkaz                                         | Příkazy řízení běhu programu                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Podmíněné řízení běhu programu                                                                    | Konečnost cyklu |
|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-----------------|
| Příklad – test zda-li je číslo prvočíslem – isPrimeNumber() 2/3 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                   |                 |
|                                                                 | <ul style="list-style-type: none"><li>Hodnota výrazu <code>(int)sqrt((double)n)</code> se v cyklu nemění.</li></ul> <pre>1 for (int i = 2; i &lt;= (int)sqrt((double)n); ++i) { 2     ... 3 }</pre> <ul style="list-style-type: none"><li>Můžeme použít <b>operátor čárky</b> a inicializovat proměnou <code>maxBound</code>.</li></ul> <pre>1 for (int i = 2, maxBound = (int)sqrt((double)n); 2     i &lt;= maxBound; ++i) { 3     ...</pre> <ul style="list-style-type: none"><li>Nebo můžeme definovat proměnnou <code>maxBound</code> jako konstantní.</li></ul> <pre>1 _Bool ret = true; 2 const int maxBound = (int)sqrt((double)n); 3 for (int i = 2; i &lt;= maxBound ; ++i) { 4     ... 5 }</pre> <p>Příklad kompilace a spuštění <code>demo-prime.c</code>: <code>clang demo-prime.c -lm; ./a.out 13.</code></p> | <p><i>Striktně vzato ji počítáme znovu a znovu při každém testu podmínky opakování cyklu.</i></p> |                 |

## Část II

## Výrazy

### Základní rozdělení operátorů

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none"><li>Můžeme rozlišit čtyři základní typy binárních operátorů:<ul style="list-style-type: none"><li>Aritmetické operátory – sčítání, odčítání, násobení, dělení;</li><li>Relační operátory – porovnání hodnot (menší, větší, ...);</li><li>Logické operátory – logický součet a součin;</li><li><b>Operátor přiřazení</b> - na levé straně operátoru = je proměnná (l-hodnota reprezentující místo v paměti).</li></ul></li><li>Unární operátory:<ul style="list-style-type: none"><li>indikující kladnou/zápornou hodnotu: + a –;<p><i>operátor – modifikuje znaménko výrazu za ním.</i></p></li><li>modifikující proměnou: ++ a --;</li><li>logický operátor doplněk: !;</li><li>bitová negace : ~ (negace bit po bitu).</li></ul></li><li>Ternární operátor – podmíněný příkaz.</li></ul> <p><i>Jediný ternární operátor v C je podmíněný příkaz ? :</i></p> <p><a href="http://www.tutorialspoint.com/cprogramming/c_operators.htm">http://www.tutorialspoint.com/cprogramming/c_operators.htm</a></p> |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

## Aritmetické operátory

- Operandy aritmetických operátorů mohou být libovolného aritmetického typu.

Výjimkou je operátor zbytek po dělení % definovaný pro celá čísla (int).

|    |               |         |                                                |
|----|---------------|---------|------------------------------------------------|
| *  | Násobení      | x * y   | Součin x a y                                   |
| /  | Dělení        | x / y   | Podíl x a y                                    |
| %  | Dělení modulo | x % y   | Zbytek po dělení x a y                         |
| +  | Sečítání      | x + y   | Součet x a y                                   |
| -  | Odčítání      | x - y   | Rozdíl a y                                     |
| +  | Kladné znam.  | +x      | Hodnota x                                      |
| -  | Záporné znam. | -x      | Hodnota -x                                     |
| ++ | Inkrementace  | ++x/x++ | Inkrementace před/po vyhodno-<br>cení výrazu x |
| -- | Dekrementace  | --x/x-- | Dekrementace před/po vyhodno-<br>cení výrazu x |

## Logické operátory

- Operandy mohou být aritmetické typy nebo ukazatele.
- Výsledek 1 má význam true, 0 má význam false.
- Ve výrazech && a || se vyhodnotí nejdříve levý operand.
- Pokud je výsledek dán levým operandem, pravý se nevyhodnocuje.

Zkrácené vyhodnocování – složité výrazy.

|    |             |        |                                                        |
|----|-------------|--------|--------------------------------------------------------|
| && | Logické AND | x && y | 1 pokud x ani y není rovno 0, ji-<br>nak 0.            |
|    | Logické OR  | x    y | 1 pokud alespoň jeden z x, y není<br>rovno 0, jinak 0. |
| !  | Logické NOT | !x     | 1 pro x rovno 0, jinak 0.                              |

- Operace && a || se vyhodnocují zkráceným způsobem, tj. druhý operand se nevyhodnocuje, pokud lze výsledek určit již z hodnoty prvního operandu.

## Operace bitového posunu

- Operátory bitového posunu posouvají celý bitový obraz o zvolený počet bitů vlevo nebo vpravo.
  - Při posunu vlevo jsou uvolněné bity zleva plněny 0.
  - Při posunu vpravo jsou uvolněné bity zprava:
    - u čísel kladných nebo typu unsigned plněny 0;
    - u záporných čísel buď plněny 0 (logický posun) nebo 1 (aritmetický posun vpravo), dle implementace překladače.
- Operátory bitového posunu mají nižší prioritu než aritmetického operátory!
  - i << 2+1 znamená i << (2+1).

Nebuďte zaskočení nečekanou interpretací – závorkujte!

## Unární aritmetické operátory

- Unární operátory ++ a -- mění hodnotu svého operandu.

Operand musí být l-hodnota, tj. výraz, který má adresu, kde je uložena hodnota výrazu (např. proměnná).

  - Lze zapsat prefixově např. ++x nebo --x;
  - nebo postfixově např. x++ nebo x--;
  - v obou případech se však liší výsledná hodnota výrazu!

| int i; int a; | hodnota i                         | hodnota a |
|---------------|-----------------------------------|-----------|
| i = 1; a = 9; | 1                                 | 9         |
| a = i++;      | 2                                 | 1         |
| a = ++i;      | 3                                 | 3         |
| a = ++(i++);  | nelze, hodnota i++ není l-hodnota |           |

V případě unárního operátoru i++ je nutné v paměti uchovat původní hodnotu i a následně inkrementovat hodnotu proměnné i. V případě použití ++i pouze inkrementujeme hodnotu i. Proto může být použití ++i efektivnější.

## Bitové operátory

- Bitové operátory vyhodnocují operandy bit po bitu.
- |    |              |        |                                                                                     |
|----|--------------|--------|-------------------------------------------------------------------------------------|
| &  | Bitové AND   | x & y  | 1 když x i y je rovno 1 (bit po bitu).                                              |
|    | Bitové OR    | x   y  | 1 když x nebo y je rovno 1 (bit po bitu).                                           |
| ^  | Bitové XOR   | x ^ y  | 1 pokud pouze x nebo pouze y je 1 (exkluzivně právě jedna z variant) (bit po bitu). |
| ~  | Bitové NOT   | ~x     | 1 pokud x je rovno 0 (bit po bitu).                                                 |
| << | Posun vlevo  | x << y | Posun x o y bitů vlevo.                                                             |
| >> | Posun vpravo | x >> y | Posun x o y bitů vpravo.                                                            |

## Operátory přístupu do paměti

Zde pro úplnost, více v následujících přednáškách.

- V C lze přímo přistupovat k adrese paměti proměnné, kde je uložena hodnota.

Potřebujeme v scanf()!

- Přístup do paměti je prostřednictvím ukazatele (pointeru).

Dává velké možnosti, ale také vyžaduje zodpovědnost.

| Operátor | Význam             | Příklad | Výsledek                                           |
|----------|--------------------|---------|----------------------------------------------------|
| &        | Adresa proměnné    | &x      | Ukazatel (pointer) na x                            |
| *        | Nepřímá adresa     | *p      | Proměnná (nebo funkce) adresovaná point-<br>erem p |
| []       | Prvek pole         | x[i]    | *(x+i) – prvek pole x s indexem i                  |
| .        | Prvek struct/union | s.x     | Prvek x struktury s                                |
| ->       | Prvek struct/union | p->x    | Prvek struktury adresovaný ukazatelem p            |

Operandem operátoru & nesmí být bitové pole a proměnná typu register.

Operátor nepřímé adresy \* umožňuje přístup na proměnné přes ukazatel.

## Relační operátory

- Operandy relačních operátorů mohou být aritmetického typu, ukazatele shodného typu nebo jeden z nich NULL nebo typ void.

|    |                  |        |                                      |
|----|------------------|--------|--------------------------------------|
| <  | Menší než        | x < y  | 1 pro x je menší než y, jinak 0.     |
| <= | Menší nebo rovno | x <= y | 1 pro x menší nebo rovno y, jinak 0. |
| >  | Větší než        | x > y  | 1 pro x je větší než y, jinak 0.     |
| >= | Větší nebo rovno | x >= y | 1 pro x větší nebo rovno y, jinak 0. |
| == | Rovná se         | x == y | 1 pro x rovno y, jinak 0.            |
| != | Nerovná se       | x != y | 1 pro x nerovno y, jinak 0.          |

## Příklad – bitových operací

```
1 #include <inttypes.h>
2
3 uint8_t a = 4;
4 uint8_t b = 5;
5
6 a      dec: 4 bin: 0100
7 b      dec: 5 bin: 0101
8 a & b  dec: 4 bin: 0100
9 a | b  dec: 5 bin: 0101
10 a ^ b  dec: 1 bin: 0001
11
12 a >> 1 dec: 2 bin: 0010
13 a << 1 dec: 8 bin: 1000
```

lec03/bits.c

Vizte rekurzivní implementaci lec03/bits-recursive.c

## Ostatní operátory

- Operandem sizeof() může být jméno typu nebo výraz.

|        |                      |           |                                                                        |
|--------|----------------------|-----------|------------------------------------------------------------------------|
| ()     | Volání funkce        | f(x)      | Volání funkce f s argumentem x                                         |
| (type) | Přetypování (cast)   | (int)x    | Změna typu x na int                                                    |
| sizeof | Velikost prvku       | sizeof(x) | Velikost x v bajtech                                                   |
| ? :    | Podmíněný příkaz     | x ? y : z | Proved' y pokud x != 0 jinak z                                         |
| ,      | Postupné vyhodnocení | x, y      | Vyhodnotí x pak y, výsledek operátoru je<br>výsledek posledního výrazu |

- Operandem operátoru sizeof() může být jméno typu nebo výraz.

```
1 int a = 10;
2 printf("%lu %lu\n", sizeof(a), sizeof(a + 1.0));
```

lec03/sizeof.c

- Příklad použití operátoru čárka.

```
1 for (c = 1, i = 0; i < 3; ++i, c += 2) {
2     printf("i: %d c: %d\n", i, c);
3 }
```

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                                       |           |                    |           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---------------------------------------|-----------|--------------------|-----------|
| Výrazy a operátory                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Přirazení | Výrazy a operátory                    | Přirazení | Výrazy a operátory | Přirazení |
| Operátor přetypování                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |           | Asociativita a prioritizace operátorů |           | Přirazení          |           |
| <div><div><div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div><div><div><div></div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div> |           |                                       |           |                    |           |

## Část V

## Appendix

### Kódovací příklad – Tisk hodnot v šestnáctkové soustavě

- Reprezentace `float` hodnot.
  - Hodnota 85.125 je **0x42aa4000**.
  - Hodnota 0.1 je sice **0x3dcccccd**, ale je kódována **0x3dcccccd**. Protože chyba je absolutně menší.
- Implementujeme funkci pro tisk paměťové reprezentace hodnoty typu `float` jako posloupnosti hodnot bajtů v šestnáctkové soustavě.
- Přístup k `float` jako posloupnosti bajtů a tisk hex hodnot "**%02x**" funkcí `printf()`.
  - Adresním operátorem `&` získáme adresu proměnné.
  - Přetypujeme adresu jako ukazatel na hodnotu `char`.
  - Použijeme nepřímý adresní operátor `*` k přístupu k hodnotě na adrese uložené v ukazateli.

### Kódovací příklad – Tisk hodnot v šestnáctkové soustavě 1/3

- Získáme adresu proměnné `float` v operátorem `&`.
- K hodnotám na adrese `&v` budeme přistupovat jako k bajtům, proto přetypujeme adresu na ukazatel (adresu) na hodnoty typu `char`.
  - `unsigned char *p = (unsigned char*)&v;`
- Hodnotu uloženou na adrese `p` získáme operátorem nepřímého adresování `*p`.
  - Adresu následujícího bajtu za adresou uloženou v `p` získáme `p = p + 1;`.
    - Protože se jedná o ukazatel na `char`, probíhá inkrementace o `sizeof(char)`, tj. o 1 (ukazatelová aritmetika).
- Vytíštěné hodnoty jsou v opačném než očekávaném pořadí **0x42aa4000** a **0x3dcccccd**.

```
1 int main(void)
2 {
3     print_float_hex(85.125);
4     print_float_hex(0.1);
5     ...
6     void print_float_hex(float v)
7     {
8         unsigned char *p = (unsigned char*)&v;
9         printf("Value %13.10f is 0x", v);
10        for (int i = 0; i < sizeof(float); ++i,
11             p = p + 1) {
12            printf("%02x", *p); // or use p[i]
13        }
14        putchar('\n');
```

```
1 $ clang floats.c -o floats && ./floats
2 Value 85.1250000000 is 0x0040aa42
3 Value 0.1000000015 is 0xcdcccc3d
```

### Nedefinované chování

- Dle standardu C mohou některé příkazy (výrazy) způsobit **nedefinované chování**.
    - `c = (b = a + 2) - (a - 1);`
    - `j = i * i++;`
  - Program se může chovat rozdílně podle použitého kompilátoru, případně nemusí jít zkompileovat, spustit, nebo dokonce padat a chovat se neobvykle či produkovat nesmyslné výsledky.
  - To se může například také stát v případě, že nejsou proměnné inicializovány.
- Vyhybejte se příkazům (výrazům), které mohou vést na nedefinované chování!**

### Přehled operátorů a jejich priorit 1/3

| Priorita | Operátor                                                                                                               | Asociativita   | Operace                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | <code>++</code><br><code>--</code><br><code>()</code><br><code>[]</code><br><code>.</code><br><code>-&gt;</code>       | P/L<br><br>L→P | pre/post inkrementace<br>pre/post dekrementace<br>volání metody<br>indexace do pole<br>přístup na položky struktury/unionu<br>přístup na položky přes ukazatel |
| 2        | <code>! ~</code><br><code>- +</code><br><code>()</code><br><code>*</code><br><code>&amp;</code><br><code>sizeof</code> | P→L            | logická a bitová negace<br>unární plus (minus)<br>přetypování<br>nepřímé adresování (dereference)<br>adresa (reference)<br>velikost                            |

### Kódovací příklad – Tisk hodnot v šestnáctkové soustavě 3/3

- Detekce uložení můžete být založena na různých principech.
  - `#include <stdint.h>`
  - `_Bool is_big_endian(void)`
- Intuitivně můžeme uložit definovanou hodnotu, která má pouze jeden bajt nenulový a ostatní nulové.
- Využijeme složeného typu `union`, ve kterém položky sdílejí paměť a umožňují nám tak různý pohled na konkrétní block paměti.
  - Definujeme celočíselnou proměnnou o čtyřech bajtech, např., `uint32_t` z knihovny `stdint.h`.
  - Nastavíme hodnotu na `0x01 00 00 00`.
  - Otestujeme první bajt paměťové reprezentace.

```
1 #include <stdint.h>
2 _Bool is_big_endian(void)
3 {
4     union {
5         uint32_t i;
6         char c[4]; // 4 is fine here as we use
7                     uint32_t
8     } e = { 0x01000000 };
9     return e.c[0];
10 }
```

### Compiler Explorer – Analýza optimalizovaného kódu

Vliv optimalizace `-O2` na výsledný kód, který obsahuje nedefinované chování, přetečení celého čísla.

```
1 int main(void)
2 {
3     int ret = 0;
4     for (int i = 2147483640; i >= 0; ++i) {
5         ret = 12;
6     }
7     return ret;
8 }
```

Assembly output (x86\_64 gcc 12.2):

```
1 main:
2     push    rbp
3     mov     rbp, rsp
4     mov     DWORD PTR [rbp-4], 0
5     mov     DWORD PTR [rbp-8], 2147483640
6     jmp     .L2
7 .L3:
8     mov     eax, DWORD PTR [rbp-8]
9     add     DWORD PTR [rbp-4], eax
10    add     DWORD PTR [rbp-8], 1
11    .L2:
12    cmp     DWORD PTR [rbp-8], 0
13    jns     .L3
14    mov     eax, DWORD PTR [rbp-4]
15    pop     rbp
16    ret
```

Output (07):

```
1 main:
2     .L2:
3     jmp     .L2
```

<https://godbolt.org/z/G3GEz4vbw>

Přehled operátorů a jejich priorit 2/3

| Priorita | Operátor                                                | Asociativita | Operace                    |
|----------|---------------------------------------------------------|--------------|----------------------------|
| 3        | <b>*</b> , <b>/</b> , <b>%</b>                          | L→R          | násobení, dělení, zbytek   |
| 4        | <b>+</b> -                                              |              | sčítání, odečítání         |
| 5        | <b>&gt;&gt;</b> , <b>&lt;&lt;</b>                       |              | bitový posun vlevo, vpravo |
| 6        | <b>&lt;</b> , <b>&gt;</b> , <b>&lt;=</b> , <b>&gt;=</b> |              | porovnání                  |
| 7        | <b>==</b> , <b>!=</b>                                   |              | rovno, nerovno             |
| 8        | <b>&amp;</b>                                            |              | bitový AND                 |
| 9        | <b>^</b>                                                |              | bitový XOR                 |
| 10       | <b> </b>                                                |              | bitový OR                  |
| 1        | <b>&amp;&amp;</b>                                       |              | logický AND                |
| 12       | <b>  </b>                                               |              | logický OR                 |

Přehled operátorů a jejich priorit 3/3

| Priorita | Operátor                                                                                                                                               | Asociativita | Operace                                                                                                                                                                 |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13       | <b>? :</b>                                                                                                                                             | P→L          | <a href="#">ternární operátor</a>                                                                                                                                       |
| 14       | <b>=</b><br><b>+=</b> , <b>-=</b><br><b>*=</b> , <b>/=</b> , <b>%=</b><br><b>&lt;&lt;=</b> , <b>&gt;&gt;=</b><br><b>&amp;=</b> , <b>^=</b> , <b> =</b> | P→L          | přirazení<br>přirazení<br>přirazení součtu, rozdílu<br>přirazení součinu, podílu a zbytku<br>přirazení bitového posunu vlevo, vpravo<br>přirazení bitového AND, XOR, OR |
| 15       | <b>,</b>                                                                                                                                               | L→P          | <a href="#">operátor čárka</a>                                                                                                                                          |

[http://en.cppreference.com/w/c/language/operator\\_precedence](http://en.cppreference.com/w/c/language/operator_precedence)