

Aritmetika a cykly

David Weber

22. prosince 2022

Připomenutí z minula

- Příkazy:
 - `printf(...)` → výpis
 - `scanf(...)` → načtení vstupu
 - `%d`, `%f`, `%s`, ... → formátové specifikace
- Větvení programu:

```
if (podmínka) {  
    // Kód programu  
}
```
- Logické spojky `&&` a `||`.

Co dnes probereme

- Datové typy
- Aritmetické operátory a knihovna `math`
- Cykly `while` a `for`

Datové typy

- **Datový typ** definuje druh hodnot, kterých smí nabývat proměnná (nebo konstanta).
- 2 druhy:
 - **jednoduché** \leftarrow typicky přímou součástí jazyka,
 - **složené** \leftarrow obsahují více prvků (stejného či různých datových typů).
- Deklarace v jazyce C: `<datový typ> <název proměnné>`.

S čím budeme pracovat...

- **int** (integer) \leftarrow celočíselný datový typ
 - 16 nebo 32 bitů (typicky 32)
- **float** (floating point) \leftarrow desetinný datový typ
 - 32 bitů
 - uchovává 6 platných cifer.
- **double** \leftarrow desetinný datový typ (dvojnásobný rozsah oproti float)
 - 64 bitů
 - uchovává 15 platných cifer.
- Přehled všech datových typů v C (pro zájemce): [zde](#)

Další datové typy

- `char` ← reprezentuje číslo nebo znak
 - 8 bitů
- Řetězce znaků `"..."` ← složitější, zatím nebudeme řešit 😊
- `void` ← prázdný datový typ, nenabývá žádné hodnoty.

Aritmetika

- Jazyk C v základu podporuje standardní aritmetické operace.
- Operátory:
 - Sčítání +
 - Odčítání -
 - Násobení *
 - Dělení /

Priorita operací je stejná jako v matematice, popř. ji lze stanovit pomocí závorek (,).

Příklad

```
#include <stdio.h>

int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);

    int sum = a + b;

    printf("%d", sum);
}
```


Příklad

Procentuální podíl x vůči y :

```
#include <stdio.h>

int main(void) {
    float x, y;
    scanf("%f %f", &x, &y);

    float percent = 100 * x / y;

    printf("%g", percent);

    return 0;
}
```

Knihovna `math`

- Standardní aritmetické operace nám již nebudou stačit, pokud budeme chtít počítat se složitějšími výrazy.
- Knihovna `math` poskytuje různé matematické funkce pro počítání např. s
 - **`mocninami`,**
 - **`odmocninami`,**
 - **`goniometrickými funkcemi`,**
 - **`logaritmy`,**
 - ...

Přehled některých funkcí

- Co se vám může hodit:
 - $\text{pow}(x, n) \leftarrow \text{vypočítá } x^n.$
 - $\text{sqrt}(x) \leftarrow \text{vypočítá } \sqrt{x}.$
 - $\text{floor}(x) \leftarrow \text{zaokrouhlení čísla } x \text{ dolů.}$
 - $\text{ceil}(x) \leftarrow \text{zaokrouhlení čísla } x \text{ nahoru.}$
- Všechny vypočtené hodnoty jsou v přesnosti **double**.
- Kompletní přehled (pro zájemce) [zde](#)

Cykly

- Často budeme potřebovat nějakou část kódu vykonat vícekrát.
- Je pochopitelně hloupost kopírovat stejný kód několikrát pod sebe. 😊
- K tomu se nám bude hodit tzv. cyklus (anglicky *loop*).
- V jazyce C existují 3 typy:
 - `While`
 - `Do While`
 - `For`

Cyklus while

- Používáme, pokud dopředu neznáme počet opakování cyklu.
- Syntaxe:

```
while (podmínka) {  
    // Kód programu  
}
```

- Odpovídá **iteraci s testem na začátku**.

Příklad

```
#include <stdio.h>
int main(void) {
    int n;
    scanf("%d", &n);
    while (n > 1) {
        print("%d", n);
        n = n / 2;
    }
    return 0;
}
```

Příklad

I zde lze použít logické spojky (jako u `if`):

```
#include <stdio.h>
int main(void) {
    int x, y;
    scanf("%d %d", &x, &y);
    while (x > 1 || y < 1000) {
        printf("x = %d\ny = %d", x, y);
        x = x / 2;
        y = y * 2;
    }
    return 0;
}
```

Cyklus for

- Používáme, pokud naopak dopředu známe počet opakování.
- Syntaxe:

```
for (deklarace; podmínka; aktualizace) {  
    // Kód  
}
```


Cyklus for

- Syntaxe:

```
for (deklarace; podmínka; aktualizace) {  
    // Kód  
}
```

- Popis činnosti:

- provede se deklarace proměnné,
- zkontroluje se podmínka,
- pokud podmínka platí, provede se tělo cyklu,
- na vykonání těla cyklu se provede **aktualizace**,
- znovu se zkontroluje podmínka a cyklus se opakuje,
- pokud podmínka neplatí, činnost cyklu končí.

Cyklus for

- Typický příklad zápisu:

```
for (int i = 0; i < 10; i++) {  
    // Kód  
}
```

- $i++$ je ekvivalent zápisu $i = i + 1$ (nejčastěji se používá právě u for cyklu).

Příklad

Součet všech sudých čísel menších než n :

```
int n;  
scanf("%d", &n);  
int sum = 0;  
for (int i = 0; i < n; i++) {  
    if (i % 2 == 0) {  
        sum += i;  
    }  
}  
printf("%d", sum);
```

Příklad

Nebo jinak:

```
int n;  
scanf("%d", &n);  
int sum = 0;  
for (int i = 0; i < n; i += 2) {  
    sum += i;  
}  
printf("%d", sum);
```

Dodatek k desetinným číslům

- Nikdy neporovnáváme desetinná čísla na rovnost ✗

```
float x = 0.1;  
x += 0.2;  
if (x == 0.3) {  
    // Kód  
}
```

- Lze pouze porovnávat max. odchylku ✓

```
if (fabs(x - 0.3) < 0.01) {  
    // Kód  
}
```

Dodatek k desetinným číslům

- **Důvod:** při počítání s desetinnými čísly dochází k zaokrouhlovací chybě.

Otázky?

