

IB001 Úvod do programování skrze C

Cvičení 2

Petr Velan

velan@mail.muni.cz

Fakulta informatiky
Masarykova univerzita

24. 9. 2012

Co bylo na přednášce

- Organizace
- Co je program, programovací jazyk
- Životopis programu: zdrojový kód, preprocesor, překladač, linker
- Debugger
- Základní pojmy
 - Identifikátor
 - Klíčové slovo
 - Řetězcová konstanta
 - Komentář

Co bylo na přednášce

- Preprocessor
 - Direktiva preprocesoru
 - *#define, #include, #if, #else, #endif, #elif, #ifdef, #ifndef*
- Překladač (kompilátor)
- Linker
 - **.a, *.lib, *.so, *.dll*

Co bylo na přednášce

- Debugger
 - Krokování
 - Sledování hodnot
 - Breakpointy
- Struktura programu
 - Direktivy preprocesoru
 - Globální deklarace a definice
 - Funkce (main)

Co bylo na přednášce

- Funkce
 - Hlavička
 - Tělo funkce
 - Návrátová hodnota (main)
 - Prototyp funkce
- Posloupnost vykonávání příkazů
 - Lineární, ovlivněna řídicími strukturami
 - Příkazy odděleny středníkem
- *puts(3), printf(3)*
- Formátování kódu

Rychlý přehled

- Cheat Sheet

- `http://www.cheat-sheets.org/saved-copy/C.Reference.Card.ANSI.2.2.pdf`
- `http://faculty.olin.edu/~jcrisman/Teaching/SigSysWeb/Embedded/c_cheat_sheet.htm`

Základní datové typy

- Datové typy
 - *char* - neměšší adresovatelná jednotka (byte)
 - *int* - základní celočíselný typ velikosti alespoň 16 bitů
 - *float* - typ pro desetinná čísla, typicky 32 bitů
 - *double* - typ pro desetinná čísla s větší přesností, typicky 64 bitů
 - Velikost *float* a *double* je závislá na implementaci.
- Typ *void* slouží pro vyjádření skutečnosti, že funkce nevrací žádnou návratovou hodnotu, nebo že nemá žádné argumenty

```
void printHello (void)
{
    printf("Hello\n");
}
```

Základní datové typy

- Volitelné specifikátory
 - *signed* - zaručuje, že daný typ bude se znaménkem (celá čísla)
 - *unsigned* - zaručuje, že daný typ bude bez znaménka (nezáporná)
 - *short* - *short int* je veliký alespoň 16 bitů, *int* není kratší než *short int*
 - *long* - *long int* není kratší než *int*, typicky 32 bitů. *long long int* není kratší než *long int*, typicky velikosti 64 bitů, od verze C99.
- $16b \leq \text{short int} \leq \text{int} \leq 32b \leq \text{long int} \leq 64b \leq \text{long long int}$
- Unární operátor *sizeof*
- http://en.wikipedia.org/wiki/C_data_types

Proměnné, čísla a výpočty

- Každá proměnná má svůj typ. Podle velikosti daného typu je pro ni alokováno místo v paměti
- Při volání funkcí se předává kopie hodnoty
- Čísla je možno zapisovat celočíselně, s desetinnou tečkou, hexadecimálně nebo v osmičkové soustavě
- C podporuje základní aritmetické operátory $+$, $-$, $*$, $/$, včetně závorek ()
- Navíc má širokou podporu pro bitové operace
- Vyhodnocení probíhá podle priority operátorů (<http://www.swansontec.com/sopc.html>)
- **Typová konverze** (<http://physics.ujep.cz/~mmaly/vyuuka/oporaPrgB/018.html>)

Proměnné, čísla a výpočty - Příklad

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

int main(void)
{
    int i = 3, j = 4, k;
    double pi = 3.14159;

    k = (i + j) * pi;
    printf("%i ", k);

    return 0;
}
```

Procvičení - typy

- Vytvořte program, který vytiskne velikosti následujících datových typů:
 - *char*
 - *short int*
 - *int*
 - *long int*
 - *long long int*
 - *float*
 - *double*
 - *long double*

Datové typy s pevnou délkou

- Někdy je potřeba předem znát velikost datových typů, kvůli kompatibilitě
- V hlavičkovém souboru *stdint.h* jsou definice následujících typů
 - *int8_t, uint8_t*
 - *int16_t, uint16_t*
 - *int32_t, uint32_t*
 - *int64_t, uint64_t*

Procvičení - výstup

- Vytvořte program pro výpočet objemu kvádrů
 - Pomocí *#define* vytvořte konstanty A, B, C
 - Uložte hodnoty konstant do proměnných typu *double*
 - Do nové proměnné uložte objem kvádrů o hranách délek A,B,C
 - Výslednou proměnnou vytiskněte pomocí funkce *printf(3)*
 - Dbejte na logické pojmenování proměnných a formátování kódu.

Procvičení - výstup II.

- Upravte předchozí program tak, aby výpočet objemu probíhal v samostatné funkci