

# Základy jazyka C

## 1. cvičení

Jiří Zacpal

KMI/ZP1 – Základy programování 1

# Doporučená literatura

- Pavel Herout: *Učebnice Jazyka C*. Kopp, 2007.
- Brian W. Kernighan, Dennis M. Ritchie: *Programovací jazyk C*. CPress 2008.
- Reek Kenneth: *Pointers on C*. Addison Wesley, 1997.
- Robert Sedgewick: *Algorithms in C*. Addison-Wesley Professional, 2001.
- Jeri R. Hanly, Elliot B. Koffman: *Problem Solving and Program Design in C*. Addison Wesley, 2006.
- Eric S. Roberts: *Programming Abstractions in C*. Addison Wesley, 1997.
- Eric S. Roberts: *The Art and Science of C*. Addison Wesley, 1994.
- Libovolné další učebnice jazyka C

# Požadavky na zápočet

- pro zápočet je potřeba získat **20 bodů**:
  - 1 bod na každém cvičení za příklad
  - 0 až 5 bodů za úkoly vyhlašované v semestru (celkem 2 úkoly)
  - 0, 3, 5 bodů za písemné práce v semestru (celkem 2 za semestr)
- individuální domluva možná

# Konzultace

- v pracovně 5.044
- každé úterý 10.00 – 12.00
- jindy po vzájemné domluvě
- email: [jiri.zacpal@upol.cz](mailto:jiri.zacpal@upol.cz)
- web: [edis.upol.cz](http://edis.upol.cz)
  - podmínky zápočtu
  - příklady
  - body

# Charakteristiky jazyka C

- Nízkoúrovňový programovací jazyk
- Platformově nezávislý (při správném použití)
- Rozšířený
- Kompilovaný (překládaný)
- Typový
- Založen na práci s ukazateli (pointery)

# Historie jazyka C

- 1972 – implementace první verze Denisem Ritchiem
- 1978 – vydání knihy *The C Programming Language* od Denise Ritchie a Briana Kernighana = **C podle K&R**
- 1988 – standard **ANSI** = 2. vydání knihy
- 1990 – standard ISO = **C90** = ANSI
- 1999 – standard **C99**

# Editory

- MS Visual Studio (MS Windows)
- Code::Blocks (multiplatformní)
- GNU Emacs (GNU/Linux)
- Xcode (Mac OS)
- a mnoho dalších (teoreticky libovolný textový editor)

# MS Visual Studio 2012

- zdarma pro studenty
- vytvoření projektu
- soubory s příponou cpp x c
- vkládání existujících souborů do projektů
- kompilace, spuštění, debugging (F5xCtrl F5)
- „odvšivení“
  - spouštění krok po kroku (F10, F11)
  - breakpoint
  - sledování stavu proměnných



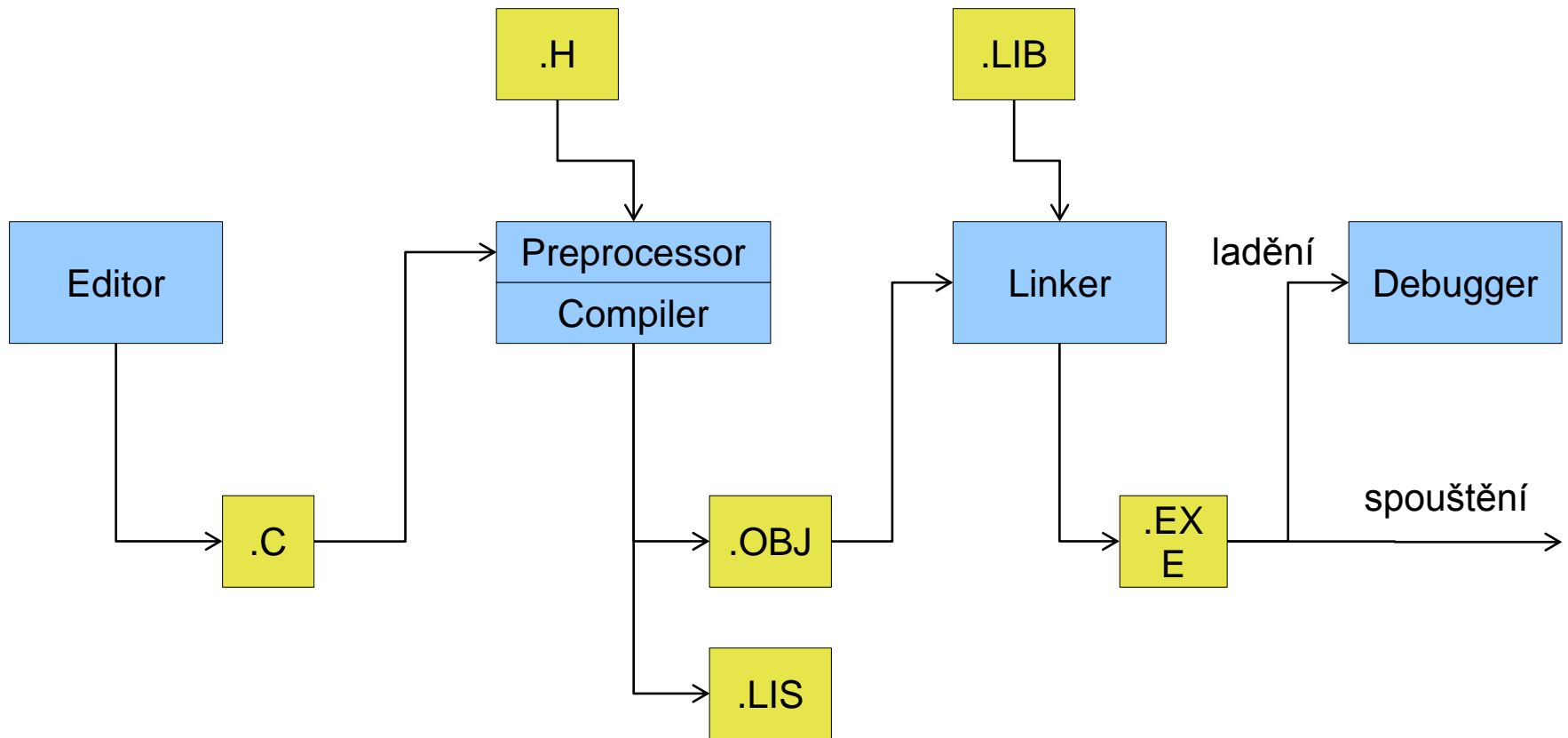
# Příklad 1

```
#include <stdio.h>
main()
{
    /* Toto je Hello world program. */
    printf("Hello world!\n");
}
/* C90 */
```

# Příklad 1

```
#include <stdio.h> //direktiva  
preprocesoru  
  
int main_1() //priklad 1  
{  
    /* Toto je Hello world program. */  
    printf("Hello world!\n");  
    return 0;  
}  
//C99
```

# Schéma vzniku programu



# Základní syntaxe

- rozlišuje velikost písmen (case-sensitive)
- ignoruje bílé znaky (odřádkování, tabulátor, mezery)
- příkazy musí být ukončeny středníkem:

```
int cislo, x, y;  
z = x*y/2/PI;  
printf("Výsledek je: %d", cislo);
```

# Základní datové typy

- Celočíselné datové typy
  - znaménkové (signed)
  - neznaménkové (unsigned)  
`char, short, int, long`
- Reálné číselné typy – s pohyblivou řádovou čárkou  
`float, double, long double`
- Textové řetězce nejsou základní datový typ

# Výstup na obrazovce

- Syntaxe obecně:

```
printf(řídící_řetězec, hodnota_1, ... );
```

- Příklady:

```
printf("Součet je %d", sum);
```

```
printf("Součet je %d", x+y);
```

```
printf("Součet je %d\t Součin je %d\n",  
x+y, x*y);
```

```
printf("Plán jsme splnili na 100%%.");
```

```
printf("Dekadicky %d je oktalově %o a  
hexadecimálně %x.\n", cislo, cislo,  
cislo);
```

# Vstup z klávesnice

- Syntaxe obecně:

```
scanf(řídící_řetězec, &promenna_1, ... );
```

- Příklady:

```
scanf("%d", &cislo);
```

```
scanf("%d %o %x", &cis1, &cis2, &cis3);
```

```
scanf("%s", muj_text);
```

# Základní možnosti řídicího řetězce

- %c – výpis nebo načtení znaku
- %d (%i) – celé číslo desítkově znaménkově
- %u – celé číslo desítkově neznaménkově
- %o – celé číslo osmičkově
- %x (%X) – celé číslo šestnáctkově (malá / velká písmena)
- %f – desetinné číslo
- %e (%E) – desetinné číslo semilogaritmicky
- %g (%G) – jako %f nebo %e (%E) podle hodnoty čísla
- %s – textový řetězec



# Příklad 2

```
main() //příklad 2
{
    int fahr, celsius;
    int dolni, horni, krok;

    dolni=0;
    horni=300;
    krok=20;

    fahr=dolni;
    while (fahr<=horni)
    {
        celsius=5*(fahr-32)/9;
        printf("%d\t%d\n", fahr, celsius);
        fahr=fahr+krok;
    }
}
```

# Příklad 2

```
main() //příklad 2
{
    float fahr, celsius;
    int dolni, horni, krok;

    dolni=0;
    horni=300;
    krok=20;

    fahr=dolni;
    while (fahr<=horni)
    {
        celsius=(5.0/9.0) * (fahr-32.0);
        printf("%3.0f\t%6.1f\n", fahr, celsius);
        fahr=fahr+krok;
    }
}
```

# Příklad 3

```
main() //priklad 3
{
    long pz;
    pz=0;
    while (getchar() !=EOF)
        ++pz;
    printf("%ld\n", pz-1);
}
//EOF - Enter-Ctrl+Z-Enter (Unix
Ctrl+D)
```

# Příklad 3

```
main() //priklad 3
{
    long pz;
    char s[100];
    scanf("%s", s);
    pz=0;
    while(s[pz] != '\0')
        ++pz;
    printf("%ld\n", pz);
}
```

# Úkol

- Vytvořte v jazyku C program, v němž budou definovány a inicializovány proměnné různých datových typů. Hodnoty proměnných pak vypište na obrazovku.
- **Příklad výstupu:**

Hodnota promenne cislo je: 3

Hodnota promenne des\_cislo je: 3.45

Hodnota promenne muj\_znak je: +

Hodnota promenne male\_cislo je: 1.2e-10

- **Povolené knihovny:** `stdio.h`, `stdlib.h`