IB001 Úvod do programování skrze C
Cvičení 4

Petr Velan
velan@mail.muni.cz

Fakulta informatiky
Masarykova univerzita

8. 10. 2012

Co bylo na přednášce

- Přiřazovací příkaz
- Úkol: Výměna obsahu dvou proměnných
  - 1) S použitím třetí proměnné
  - 2) Bez použití dalších proměnných

```
int a = 5, b = 9, tmp;

tmp = a;
a = b;
b = tmp;
```

Co bylo na přednášce

- Přiřazovací příkaz
- Úkol: Výměna obsahu dvou proměnných
  - 1) S použitím třetí proměnné
  - 2) Bez použití dalších proměnných

```
int a = 5, b = 9, tmp;

tmp = a;
a = b;
b = tmp;
```

Co bylo na přednášce

- Složený příkaz (blok kódu)
- Platnost proměnné

```
{
    int i = 5;
    i *= 2;
    printf("%d\n", i);
}
/* dalsi kod */
{
    int i = 8;
}
```

Co bylo na přednášce

- Relační operátory (porovnání)
- $<$, $>$, $<=$, $>=$, $==$, $!=$
- Menší priorita než aritmetické operátory
- $==$ a $!=$ mají menší prioritu než $>$, $<$, $<=$, $>=$
- Procvičte si psaní speciálních symbolů na české nebo anglické klávesnici.
- Úkol: Program který načte dvě čísla ze vstupu, vytiskne číslo 0 právě když se rovnají.

```
int i = 5 > 9;
printf("%d\n", i);
printf("%d\n", 4 == 2 + 2);
```

Co bylo na přednášce

- Logické operátory
- - logický součin (obě strany musí být pravdivé, aby výsledek byl pravda)
- || - logický součet (alespoň jedna strana musí být pravdivá, aby byl výsledek pravda)
- Pravda v C - číslo různé od 0

```
printf("&&: %d, %d, %d\n", 0 && 0, 1 && 1, 1 && 0);
printf("||: %d, %d, %d\n", 0 || 0, 1 || 1, 1 ||| 0);
```

Co bylo na přednášce

- Podmíněný příkaz
  - Úplný, včetně else větve
  - Neúplný, bez else větve
- Doporučuje se pro přehlednost vždycky používat blok příkazů
- V else větvi je možné použít další if
- Úkol: Program načtě celé číslo, vytiskne *“sudý”* nebo *“lichý”*
- Úkol: Načtěte dvě čísla, vytiskněte $=$, $<$ nebo $>$ podle jich vztahu

```
if (promenna == 42) {
    printf("Promenna ma hodnotu 42\n");
} else {
    printf("Promenna nema hodnotu 42\n");
}
```

Co bylo na přednášce

- Cyklus while, do-while
- Podmínka se kontroluje před začátkem vykonání bloku (while), nebo na konci (do-while)
- Je možné uvést prázdné tělo cyklu, jeden příkaz, nebo blok příkazů

```
double mocnina(double zaklad, unsigned int exponent)
{
    double tmp = zaklad;
    while (exponent > 1) {
        tmp *= zaklad;
        exponent--;
    }
    return tmp;
}
```

Co bylo na přednášce

Cyklus while, do-while

- Úkol: Napište funkci pro celočíselnou druhou odmocninu (celočíselná druhá odmocnina čísla x je takové přirozené číslo n, pro které platí $n^2 \leq x$)

Co bylo na přednášce

- Cyklus for
- Inicializace, test, aktualizace
- V c99 je možné deklarovat řídící proměnnou v inicializační části
- Přepište funkci pro mocninu pomocí for cyklu

```
double mocnina(double zaklad, unsigned int exponent)
{
    double tmp;
    for (tmp = zaklad; exponent > 1; exponent--)
        tmp *= zaklad;
    return tmp;
}
```

Co bylo na přednášce

- Cyklus for
- Inicializace, test, aktualizace
- V c99 je možné deklarovat řídící proměnnou v inicializační části
- Přepište funkci pro mocninu pomocí for cyklu

```
double mocnina(double zaklad, unsigned int exponent)
{
    double tmp;
    for (tmp = zaklad; exponent > 1; exponent--)
        tmp *= zaklad;
    return tmp;
}
```

getchar

- Funkce *getchar(3)* slouží pro přečtení jednoho znaku.
- Vrací int obsahující ASCII hodnotu načteného znaku, nebo EOF (konstanta, end of file)

Domácí úkol

- Napište program počítající počet znaků na vstupu
- Na výstupu bude: Počet malých písmen, počet velkých písmen, počet číslic v zadaném textu
- Návod: Využite skutečnosti, že kódy písmen v ASCII rostou v abecedním pořádku, použite dualitu typu char (znak, číslo)
- Odevzdejte do odevzdávárny pro svou skupinu do půlnoci pondělí 15.10.2012
- Váš program musí být možné kompilovat bez chyb a varování při nastavení překladače z první přednášky