UPR
Cvičení 1

Informace

- Cíl předmětu
  - naučit se základy programování v jazyce C
- Bodování
  - Úkoly na cvičeních - 35 bodů
  - Domácí úkoly – 20 bodů
  - Závěrečný test – 45 bodů
- http://mrl.cs.vsb.cz/people/holusa/upr_course

Kde lze C využít?

- systémové programování
- jádro operačního systému, ovladače
- mikrokontrolery
- herní engine
- high performance aplikace

První program

- Použijeme editor Visual Studio Code
- Vytvoříme/otevřeme soubor main.c

```
#include <stdio.h>
int main() {
  printf("Hello world!\n");
 return 0;
}
```

vložení hlavičkového souboru /usr/include/stdio.h

definice hlavní funkce programu, která vrací číslo (int)

zavolání funkce, která naformátuje řetězec a vytiskne na výstup

funkce vrací hodnotu 0

”na bílých znacích nezáleží”

- Přeložíme pomocí příkazového řádku

```
$ gcc main.c -o main
```

- Spustíme

```
$ ./main
```

Co se děje při překladu?

- pro každý .c soubor
  - proběhne preprocesor a zpracuje direktivy (začínají #)
    - vložení souboru <stdio.h>
    - náhrada konstant
  - zkompilováni do object file .o
- slinkování všech .o do spustitelného souboru

Proměnné

- odkazují na kus paměti pod naším jménem
- Musí mít datový typ (určuje velikost v paměti, funkce `sizeof`)

  char (1 bajt)

  int (4 bajty)

  long long (8 bajtů)

  double (8 bajtů)

Proměnné

- Syntaxe

```
<typ> <název>[ = <výraz>];
char znak = 'D';
int a = 200;
int cislo = a * (a + 1);
double pi = 3.1415926;
```

- je vhodné proměnné inicializovat
  - jazyk C nenuluje proměnné

Funkce printf

- Tiskne formátovaný řetězec na standardní výstup
- Typy formatovacího řetezce
  - %d (celé číslo), %x (hex číslo), %f (desetinné číslo), %c (znak), %s (řetězec)
- Escape znaky
  - \n (nový řádek), \r (návrat na začátek řádku), \t (tabulátor)

```
int cislo = 20;
char znak = 'D';
printf("%d,hex: %x\nznak: %c", cislo, cislo, znak);
```

Funkce scanf

- Načte data z textového vstupu do proměnných
- Je potřeba předat adresu proměnné
  - jinak nebude vědět kam to uložit
  - operátor reference & před proměnnou

```
int a, b;
printf("hodnota: %d, adresa: %p\n", a, &a);
scanf("%d %d", &a, &b);
printf("soucet %d + %d = %d\n", a, b, a + b);
```

Výrazy

- Vrací hodnotu
  - Výraz lze složit pomocí operací
    - aritmetické

      `-a, a + b, a - b, a * b, a / b, a % b`
    - porovnávací (vrací TRUE/FALSE)

      `a < b, b > a, a <= b, a >= b`

      `a == b, a != b`
    - logické (vrací TRUE/FALSE)

      NOT: `!active`

      OR: `height > 171 || age > 18`

      AND: `height > 171 && age > 18`

Komentáře

- Můžete si dělat v kódu poznámky, aniž by to ovlivnilo funkčnost programu

```
int a = 5; // do proměnné a si uložíme 5

/* taky můžeme vytvořit komentář na více
řádku */
```

Podmínky - if

```
if(<výraz>) {
<kód pokud je výraz pravdivý>
}
```

```
if(a < 0) {
  a = a * -1;
}
```

Podmínky - else

```
if(<výraz>) {
   <kód pokud je výraz pravdivý>
}
else {
   <kód pokud není výraz pravdivý>
}
```

Podmínky – více podmínek

```
if(hours >= 0 && hours < 12) {
  printf("dopoledne\n");
} else if(hours == 12) {
  printf("poledne\n");
} else if(hours <= 24) {
  printf("odpoledne\n");
} else {
  printf("chyba\n");
}
```

Blok / Scope

- Lokální proměnné existují pouze uvnitř bloku ve složených závorkách

```
int main() {
  int a = 10;
  if(a >= 10) {
    int b;      //existuje a, b
    {
      int c;   // existuje a, b, c
    }
    // existuje a, b
  }
  // existuje a
}
```

Cykly

- opakuj blok kódu dokud platí <výraz>

```
while(<výraz>) {
  <blok kódu>
}
```

```
int n = 0;
while (n < 10) {
  printf("%d\n", n);
  n = n + 1;
}
```

Úkoly

1. Vypište pomocí cyklu sudá čísla od 10 do 0
    - podmínka, zbytek po dělení (modulo - %)
2. Spočítejte sumu všech čísel a na konci vytiskněte
3. Načtěte číslo od uživatele a vypište, zda se jedná o prvočíslo

Funkce

Pojmenovaný blok kódu, který lze parametrizovat

- může vracet hodnotu (výjimkou je typ void, který nic nevrací)

Syntaxe:

```
<návratový typ> <název>(<typ1> <prom1>, ...) {
          <kód funkce>

}
```

Funkce

```
int secti(int a, int b) {
  return a + b;
}
```

funkci pak můžeme zavolat

```
int x = 10;
int suma = secti(x, 20 + x);
printf("Vysledek je %d\n", suma);
```

Úkoly

- Upravte předchozí úkol tak, že kód přesunete do funkce vracející sumu
- Parametrem funkce bude hodnota, od které se bude počítat
- V mainu zavolejte funkci a vytiskněte sumu