

# Informační a komunikační technologie

# Adresy a ukazatele v C

David Weber

Kabinet K13

[weber3@spsejecna.cz](mailto:weber3@spsejecna.cz)

# Připomenutí

- Jaký je rozdíl mezi *adresou* a *hodnotou* proměnné?
  - **Hodnota** – hodnota uložená v paměťové buňce proměnné
  - **Adresa** – číslo paměťové buňky proměnné
- Pro výpis jsme používali znak &.



- Předpona 0x značí číslo zapsané **hexadecimálně** (v C, ale i jiných jazycích).

# Příklad na úvod I

Mějme program obsahující funkci pro prohození hodnot proměnných `u` a `v`.

```
void swap(int u, int v) {  
    int temp = u;  
    u = v;  
    v = temp;  
}  
  
int main(void) {  
    int a = 5;  
    int b = 10;  
    swap(a, b);  
    printf("Hodnoty a, b: %d, %d", a, b);  
    return 0;  
}
```

# Příklad na úvod II

Jaký bude výstup předešlého programu?

- (a) Hodnoty a, b: 5, 10
- (b) Hodnoty a, b: 10, 5

# Příklad na úvod II

Jaký bude výstup předešlého programu?

(a) Hodnoty a, b: 5, 10

(b) Hodnoty a, b: 10, 5

# Příklad na úvod II

Jaký bude výstup předešlého programu?

(a) Hodnoty a, b: 5, 10

(b) Hodnoty a, b: 10, 5

⇒ nijak jsme si nepomohli 😞

# V čem je problém?

- Parametry funkci předáváme tzv. **hodnotou**.
  - Hodnoty proměnných *a* a *b* jsou zkopírovány a při volání funkce `swap` jsou nově *na zásobníku* deklarovány proměnné *u* a *v*.
    - $\Rightarrow$  funkce v konečném důsledku prohodí hodnoty proměnných *u* a *v*, nikoliv *a* a *b*
- $\Rightarrow$  mohli bychom vyřešit předáním “odkazů” na původní proměnné.

# Ukazatel (pointer)

- **Datový typ** uchovávající adresu v paměti **určitého datového typu** (existují i generické ukazatele, ale ty nebudeme řešit 😊).
- Při deklaraci je třeba uvést datový typ (tím kompilátoru říkáme, jak se má interpretovat místo v paměti, kam ukazuje), jehož adresu ukazatel uchovává, a znak \* (pro odlišení od deklarace standardní proměnné).
- Jedná se vždy o **kladné celé číslo** (formátová specifikace %p).

```
int main(void) {  
    int x = 50;  
    int *px = &x;  
    printf("Hodnota x: %d\nAdresa x: %p", x, px);  
    return 0;  
}
```



# Dereference

- Ekvivalentně lze deklaraci ukazatele zápisem `int* p` (mezera mezi `*` a `p`).
- Co když máme adresu, ale ne samotnou proměnnou?  $\Rightarrow$  **využijeme tzv. operátor dereference `*`.**
  - Umožňuje nám odkázat se přímo na **hodnotu** v paměťové buňce ležící na **adrese ukazatele**.

```
int main(void) {  
    int i = 50;  
    int *p = &i;  
    *p = 100;           // same as i = 100;  
    printf("%d", i);    // prints out "100"  
}
```

# Otázka

Jak bychom měli upravit původní program, aby fungoval?

```
void swap(int u, int v) {  
    int temp = u;  
    u = v;  
    v = temp;  
}  
  
int main(void) {  
    int a = 5;  
    int b = 10;  
    swap(a, b);  
    printf("Hodnoty a, b: %d, %d", a, b);  
    return 0;  
}
```

# Otázky?

