



KATEDRA  
INFORMATIKY  
UNIVERZITA PALACKÉHO V OLOMOUCI

# Paměť 2

## Základy programování 2

Mgr. Markéta Trnečková, Ph.D.

# Paměť

## ■ předávání argumentů funkci:

- hodnotou
- odkazem = předáváme jeho adresu

### Příklad

Jak jsou funkci předávány struktury?

# Paměť

## Příklad

```
// navratovy typ funkce: adresa typu int
int deleni(int a, int b, int *r)
{
    /* r – zbytek po deleni */
    *r = a%b;
    return a/b;
}

int main()
{
    int x,y;
    /* x = 2, y = 3*/
    x = deleni(13,5,&y);
    return 0;
}
```

# Paměť

## Příklad

Naprogramujte funkci, která bere jako argument dvě proměnné a provede výměnu hodnot těchto dvou proměnných.

## Další funkce pro práci s pamětí

- `#include <string.h>`
- `void *memcpy(void *mem1, void *mem2, size_t n);`

### Příklad

```
/* špatne. kopirujeme pouze adresu */  
foo = bar;  
  
/* kopie cyklem */  
for(int i = 0; i < m; i = i+1)  
    foo[i] = bar[i];  
  
/* kopie pomoci memcpy */  
memcpy(foo , bar , sizeof(int)*m);
```

## Další funkce pro práci s pamětí

- `#include <string.h>`
- `void *memmove(void *mem1, void *mem2, size_t n);`

### Příklad

```
/* cyklem */  
for(int i = 1; i < m; i = i+1)  
    foo[i-1] = foo[i];  
  
/* pomoci memmove */  
memmove(foo, foo+1, sizeof(int)*(m-1));
```

## Další funkce s pamětí

- `#include <string.h>`
- `void *memset(void *mem, int c, size_t n);`

### Příklad

```
/* cyklem */  
for(int i = 0; i < m; i = i+1)  
    foo[i] = 0;  
  
/* pomoci memset */  
memset(foo, 0, sizeof(int)*m);
```

### Příklad

Doprogramujte funkci na výpis pole.

Vyzkoušejte pomocí `memset()` nastavit `foo` hodnoty 1. Proč to nefunguje?

Vyzkoušejte také -1 a vysvětlete proč to funguje.

## Cvičení

- 1 Naprogramujte funkci pro spojení dvou textových řetězců. Argumentem funkce jsou dva textové řetězce ke spojení, funkce vrátí nový řetězec (jako ukazatel).
- 2 Naprogramujte funkci, která jako argument bere pole čísel a vrátí pole jejich druhých mocnin (tedy prvek na indexu  $i$  vráceného pole bude mít hodnotu druhé mocniny prvku na indexu  $i$  vstupního pole).
- 3 Napište funkci, která pro 2 zadaná čísla vrátí, zda je možné udělat jejich podíl a pokud ano, vrátí i jejich podíl.
- 4 Napište funkci `int porovnej`, která porovná 2 předané textové řetězce a vrátí `-1`, pokud je první řetězec menší než druhý, `0`, pokud jsou řetězce shodné, nebo `1`, pokud je druhý řetězec menší než první. Při práci s textovými řetězci používejte výhradně ukazatele, operátor dereference a pointerovou aritmetiku.



## Cvičení

- 6 (Úloha za 5 bodů, pokud řešení odevzdáte přes MS Teams nebo zašlete na [marketa.trneckova@gmail.com](mailto:marketa.trneckova@gmail.com) do dalšího cvičení) Napište funkci s hlavičkou `int *najdi(char *text, char *podretezec, int *pocet_vyskytu)`. Funkce v daném textovém řetězci `text` vyhledá všechny výskyty zadaného podřetězce `podretezec`. Funkce vrátí pole indexů prvních znaků nalezených podřetězců, v případě, že se v řetězci `podretezec` nevyskytuje, vrátí `NULL`. Do proměnné `pocet_vyskytu` uloží počet výskytů podřetězce v řetězci. Pro řetězec "Máma mele maso." a podřetězec "ma" vrátí dvouprvkové pole s prvky 2 a 10.