

Informatika 3

3

Smerníky 2

Smerník na funkciu

- môžeme pracovať ako s hocijakým iným smerníkom
- môže ukazovať iba na funkciu s rovnakým počtom a typmi parametrov ako bol deklarovaný
- Najčastejšie použitie:
 - implementácia univerzálnych algoritmov (napr. sort)
 - spätné volanie s funkcie (callback)
 - ošetrovanie chybových stavov (`_new_handler`)

Smerník na funkciu

- Definícia:

```
typ (*f)(typ1 p1, typ2 p2);
```

- Príklad:

```
int (*f1ptr)();
```

```
int *(*f2ptr)(int, int);
```

```
int (*f3[10])();
```

```
int *(*f4[10])(char, char);
```

```
typedef int *(*f5[10])(char, char);
```

Práca so smerníkom na funkciu

- Priradenie hodnoty smerníka

```
double (*f)(double x);
```

```
f=&sin;           // áno
```

```
f=sin;           // áno
```

```
f=&sin(3);        // takto nie !!!
```

- Volanie funkcie cez smerník

```
double (*f)(double x);
```

```
f=&sin;
```

```
double y;
```

```
y=(*f)(3.14);     // áno
```

```
y=f(3.14);        // áno
```

Smerník na funkciu ako parameter funkcie

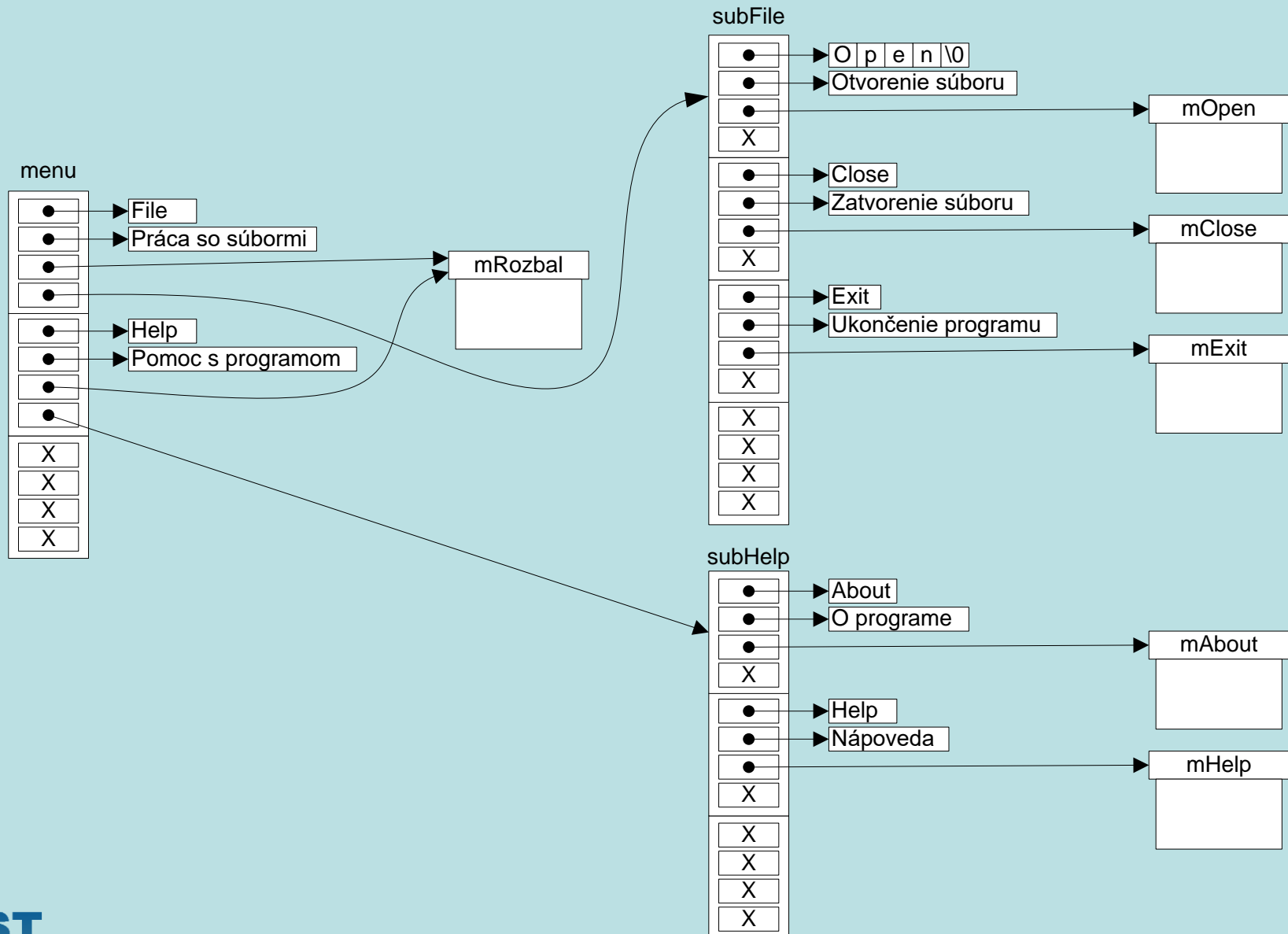
```
void Vymen(char* ptr1, char* ptr2, int velkostPolozky)
{
    char pom;
    for(char* max=ptr1+velkostPolozky ; ptr1<max ; ptr1++,ptr2++){
        pom=*ptr1;
        *ptr1=*ptr2;
        *ptr2=pom;
    }
}
```

```
void _sort(void* data, int pocet, int velkostPolozky,
          bool (*compare)(void*,void*))
{
    char* dd=(char*)data;
    char* max=dd+pocet*velkostPolozky;
    for(char* p1=dd ; p1<max ; p1+=velkostPolozky){
        for(char* p2=p1 ; p2<max ; p2+=velkostPolozky){
            if(compare(p1,p2))
                Vymen(p1,p2,velkostPolozky);
        }
    }
}
```

Smerník na funkciu ako parameter funkcie

```
bool compareIntUp(const void* x1, const void* x2)
{
    return *(int*)x1 > *(int*)x2;
}
bool compareIntDown(const void* x1, const void* x2)
{
    return *(int*)x1 < *(int*)x2;
}
void Print(int* data, int pocet)
{
    for(int i=0;i<pocet;i++) {
        printf("%i,",data[i]);
    }
    printf("\n");
}
int main()
{
    int pole[ ]={2,4,5453,5,46,56,25,6,352,31,43,56,78,4321,4,15,56,546,3};
    Print(pole,sizeof(pole)/sizeof(pole[0]));
    qsort(pole, sizeof(pole)/sizeof(pole[0]), sizeof(int), compareIntUp);
    Print(pole,sizeof(pole)/sizeof(pole[0]));
    return 0;
}
```

Použitie smerníka na funkciu - menu



const a smerníky

- smerník na znak

```
char * s;  
*s='a'; // ok  
s++;    // ok
```

- smerník na konštantný znak

```
const char * s;  
*s='a'; // nie  
s++;    // ok
```

- konštantný smerník na znak

```
char const * s;  
*s='a'; // ok  
s++;    // nie
```

- konštantný smerník na konštantný znak

```
const char const * s;  
*s='a'; // nie  
s++;    // nie
```

- príklady použitia

```
strcpy(char* dst, const char* src);  
strlen(const char* s);
```


Smerníky a argumenty funkcií

- V jazyku C++ sa odovzdávajú parametre funkciám hodnotou
- Nemáme žiadny priamy spôsob, ako vytvoriť funkciu, ktorá by:
 - menila obsah premennej vo volajúcej funkcii
 - vracala viac ako jednu hodnotu
- Na takúto zmenu sa používajú smerníky
- Ako parameter funkcii dáme smerník na premennú
- Smerník je odovzdaný hodnotou ale ukazuje na pamäť ako pôvodný smerník

Smerníky a argumenty funkcií

- Napíšte funkciu, ktorá vymení medzi sebou dva parametre

```
void vymena1(byte a, byte b)
{
    // nesprávne
    byte pom = a; a = b; b = pom;
}
```

```
void vymena2(byte *a, byte *b)
{
    // správne
    byte pom = *a; *a = *b; *b = pom;
}
```

```
void vymena3(byte &a, byte &b)
{
    // správne
    byte pom = a; a = b; b = pom;
}
```

- volanie:

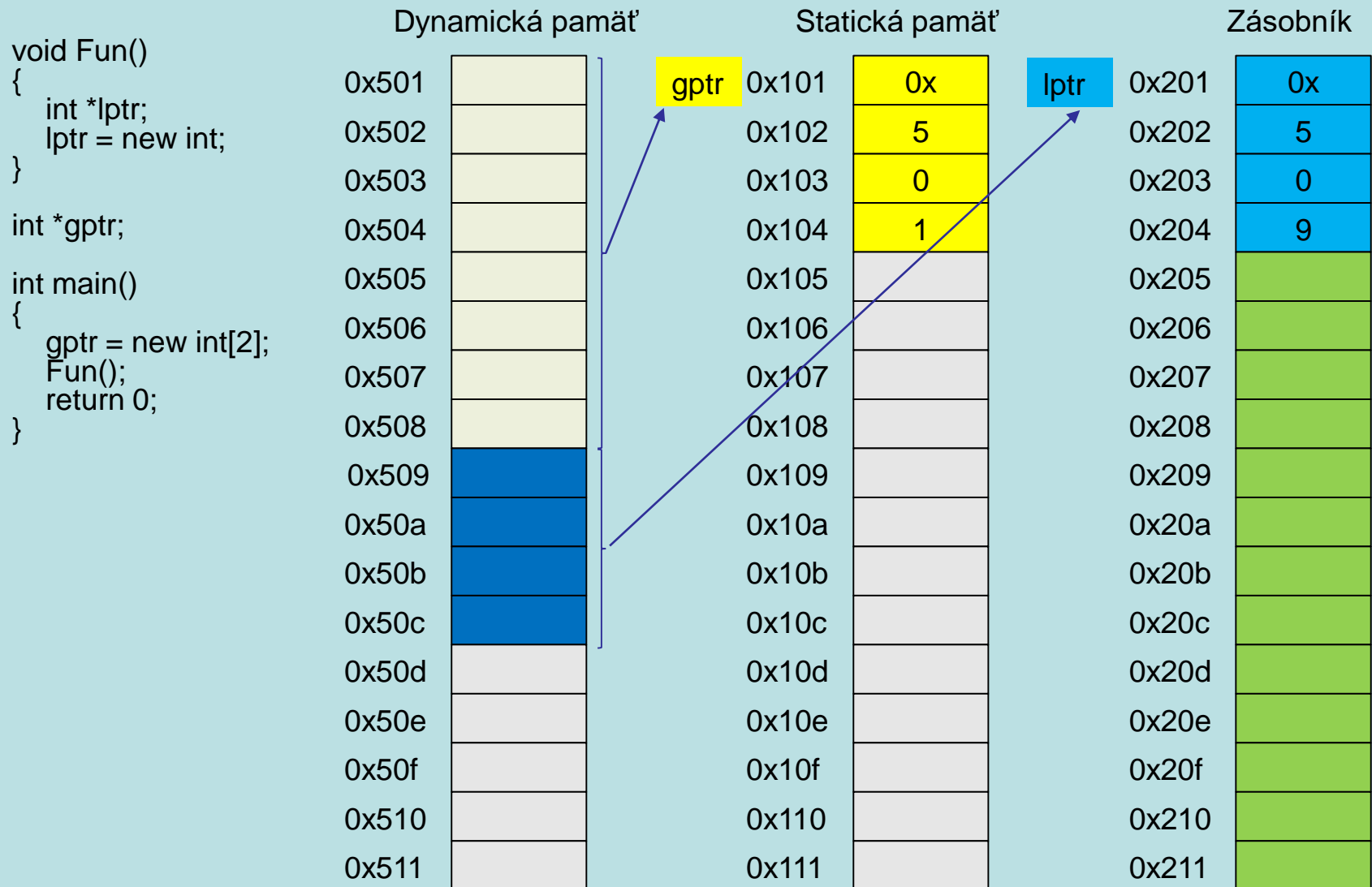
```
byte a=1, b=2;
```

```
int main()
{
    vymena1(a,b);    // na tomto mieste je v a=1 a v b=2
    vymena2(&a, &b); // na tomto mieste je v a=2 a v b=1
    vymena3(a,b);    // na tomto mieste je v a=2 a v b=1
    return 0;
}
```

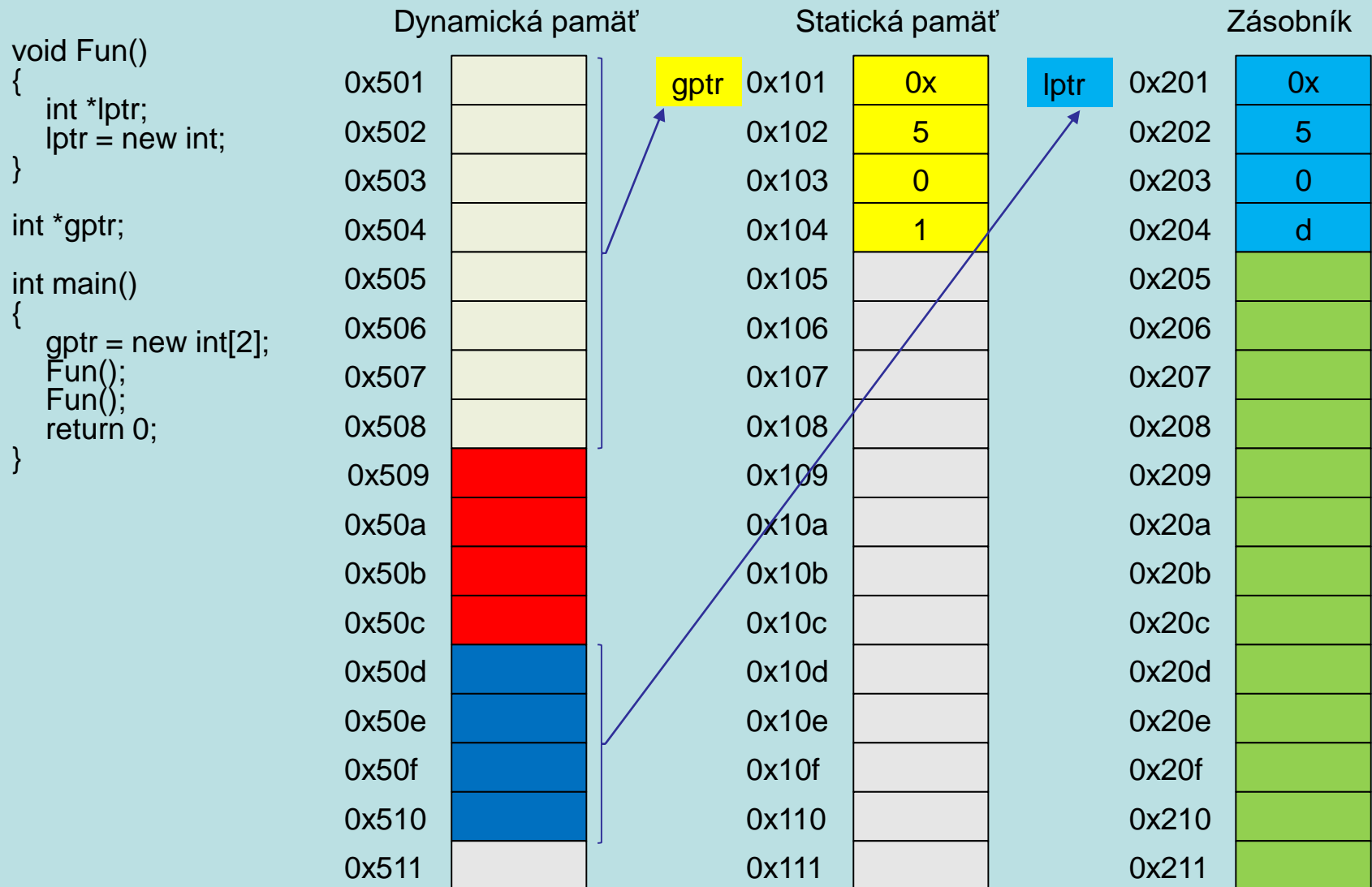
Smerníky a dynamická pamäť

| | Dynamická pamäť | | | Statická pamäť | | | Zásobník | |
|--|-----------------|--|------|----------------|---|--|----------|--|
| <pre>void Fun() { int *lptr; } int *gptr; int main() { return 0; }</pre> | 0x5001 | | gptr | 0x101 | 0 | | 0x201 | |
| | 0x5002 | | | 0x102 | 0 | | 0x202 | |
| | 0x5003 | | | 0x103 | 0 | | 0x203 | |
| | 0x5004 | | | 0x104 | 0 | | 0x204 | |
| | 0x5005 | | | 0x105 | | | 0x205 | |
| | 0x5006 | | | 0x106 | | | 0x206 | |
| | 0x5007 | | | 0x107 | | | 0x207 | |
| | 0x5008 | | | 0x108 | | | 0x208 | |
| | 0x5009 | | | 0x109 | | | 0x209 | |
| | 0x500a | | | 0x10a | | | 0x20a | |
| | 0x500b | | | 0x10b | | | 0x20b | |
| | 0x500c | | | 0x10c | | | 0x20c | |
| | 0x500d | | | 0x10d | | | 0x20d | |
| | 0x500e | | | 0x10e | | | 0x20e | |
| | 0x500f | | | 0x10f | | | 0x20f | |
| | 0x5010 | | | 0x110 | | | 0x210 | |
| | 0x5011 | | | 0x111 | | | 0x211 | |

Smerníky a dynamická pamäť



Smerníky a dynamická pamäť

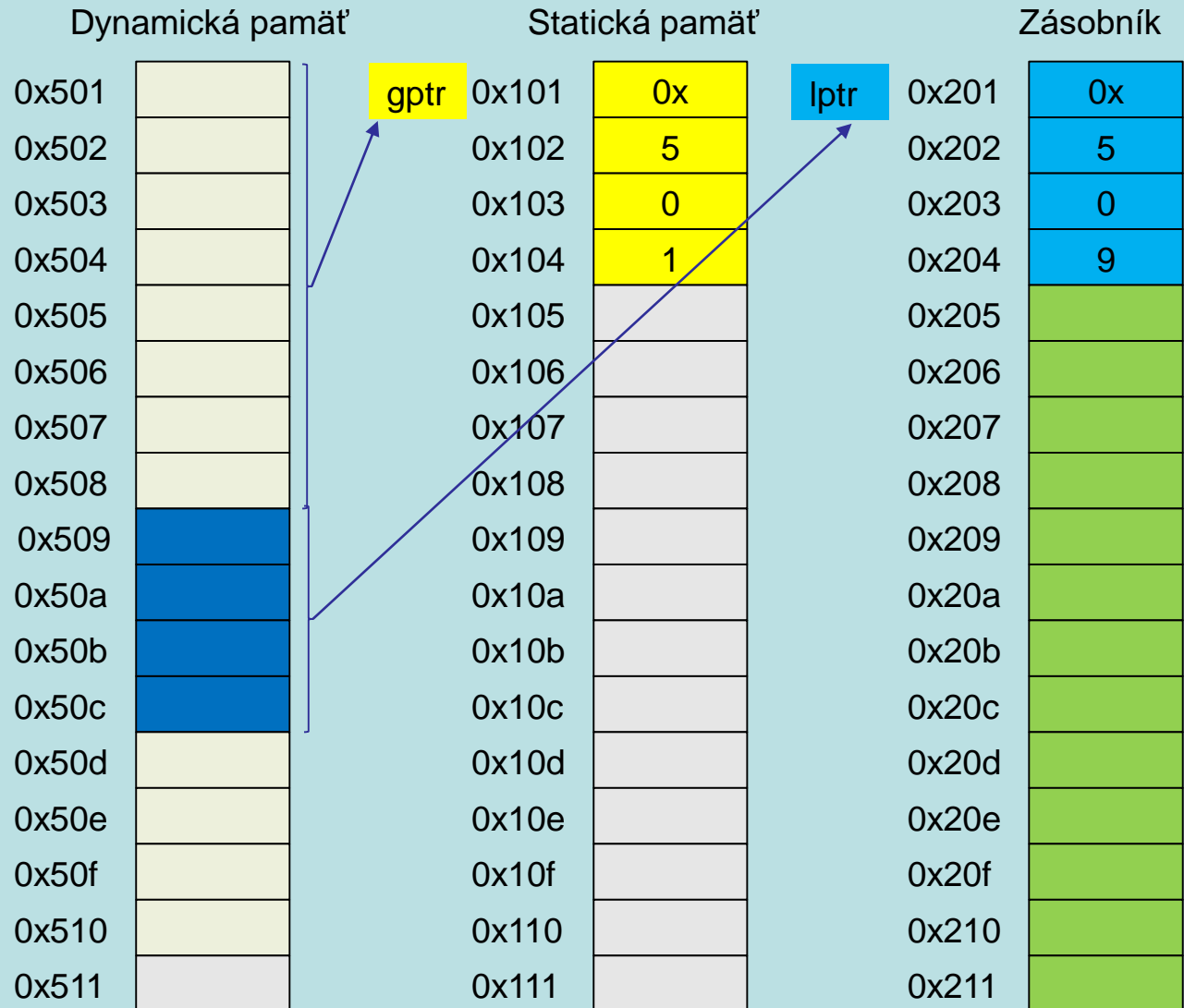


Smerníky a dynamická pamäť

```
void Fun()
{
    int *lptr;
    lptr = new int;
    // kod
    delete lptr;
}

int *gptr;

int main()
{
    gptr = new int[2];
    Fun();
    Fun();
    delete[] gptr;
    return 0;
}
```



Prenos parametrov hodnotou - 1

```
void vymena1(byte a, byte b)
{
    byte pom = a;
    a = b;
    b = pom;
}

int a=1, b=2;

int main()
{
    vymena1(a,b);
    return 0;
}
```

Statická paměť

| | |
|-------|---|
| 0x101 | 1 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |

a → a' 0x201
b → b' 0x202
pom 0x203

Zásobník

| | |
|-------|---|
| 0x201 | 1 |
| 0x202 | 2 |
| 0x203 | 1 |
| 0x204 | |
| 0x205 | |
| 0x206 | |
| 0x207 | |
| 0x208 | |
| 0x209 | |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |



Prenos parametrov hodnotou - 2

```
void vymena1(byte a, byte b)
{
    byte pom = a;
    a = b;
    b = pom;
}

int a=1, b=2;

int main()
{
    vymena1(a,b);
    return 0;
}
```

Statická paměť

| | |
|-------|---|
| 0x101 | 1 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |

Zásobník

| | |
|--------------|---|
| a → a' 0x201 | 2 |
| b → b' 0x202 | 2 |
| pom 0x203 | 1 |
| 0x204 | |
| 0x205 | |
| 0x206 | |
| 0x207 | |
| 0x208 | |
| 0x209 | |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |



Prenos parametrov hodnotou - 3

```
void vymena1(byte a, byte b)
{
    byte pom = a;
    a = b;
    b = pom;
}

int a=1, b=2;

int main()
{
    vymena1(a,b);
    return 0;
}
```

Statická paměť

| | |
|-------|---|
| 0x101 | 1 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |

a → a'
b → b'
pom

Zásobník

| | |
|-------|---|
| 0x201 | 2 |
| 0x202 | 1 |
| 0x203 | 1 |
| 0x204 | |
| 0x205 | |
| 0x206 | |
| 0x207 | |
| 0x208 | |
| 0x209 | |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |



Prenos parametrov smerníkom - 1

```
void vymena2(byte *a, byte *b)
{
    byte pom = *a;
    *a = *b;
    *b = pom;
}

byte a=1, b=2;

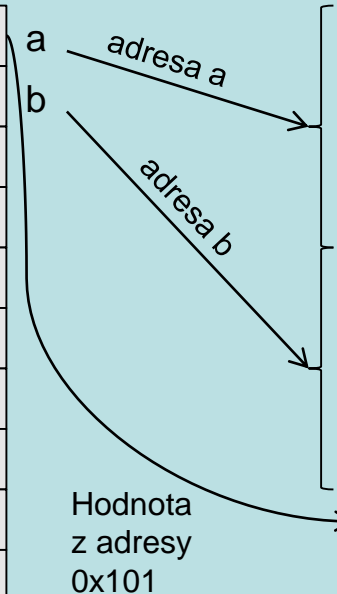
int main()
{
    vymena2(&a,&b);
}
```

Statická pamäť

| | |
|-------|---|
| 0x101 | 1 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |

Zásobník

| | |
|-------|----|
| 0x201 | 0x |
| 0x202 | 1 |
| 0x203 | 0 |
| 0x204 | 1 |
| 0x205 | 0x |
| 0x206 | 1 |
| 0x207 | 0 |
| 0x208 | 2 |
| 0x209 | 1 |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |



pom

Prenos parametrov smerníkom - 2

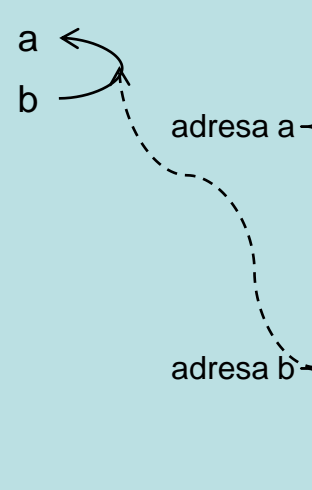
```
void vymena2(byte *a, byte *b)
{
    byte pom = *a;
    *a = *b;
    *b = pom;
}

byte a=1, b=2;

int main()
{
    vymena2(&a,&b);
}
```

Statická pamäť

| | |
|-------|---|
| 0x101 | 2 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |



Hodnota z adresy 0x101

Zásobník

| | |
|-------|----|
| 0x201 | 0x |
| 0x202 | 1 |
| 0x203 | 0 |
| 0x204 | 1 |
| 0x205 | 0x |
| 0x206 | 1 |
| 0x207 | 0 |
| 0x208 | 2 |
| 0x209 | 1 |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |

pom

Prenos parametrov smerníkom - 3

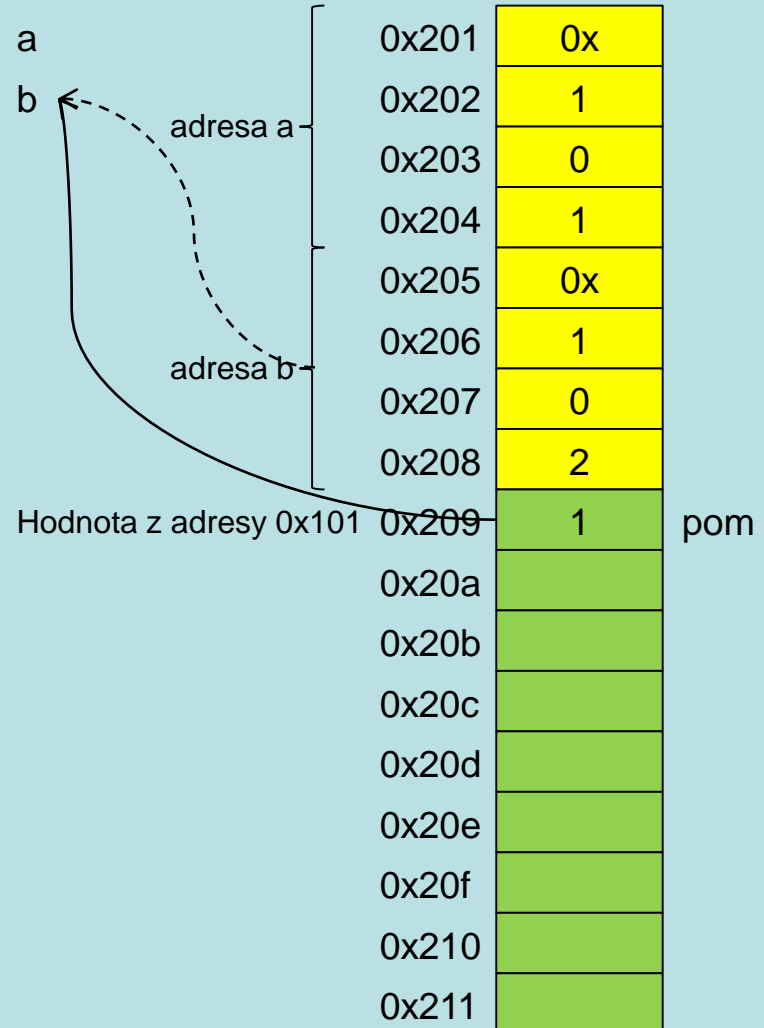
```
void vymena2(byte *a, byte *b)
{
    byte pom = *a;
    *a = *b;
    *b = pom;
}

byte a=1, b=2;

int main()
{
    vymena2(&a,&b);
}
```

Statická pamäť

| | |
|-------|---|
| 0x101 | 2 |
| 0x102 | 1 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |



Prenos parametrov odkazom - 1

```
void vymena3(byte &a, byte &b)
{
    byte pom = a;
    a = b;
    b = pom;
}

byte a=1, b=2;

int main()
{
    vymena3(a,b);
}
```

```
void vymena2(byte &a, byte &b)
{
    byte pom = &a;
    &a = &b;
    &b = pom;
}

byte a=1, b=2;

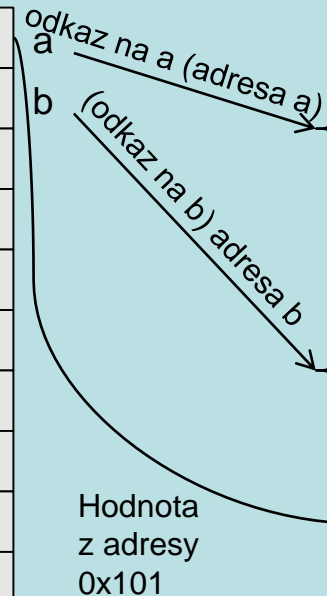
int main()
{
    vymena2(&a,&b);
}
```

Statická pamäť

| | |
|-------|---|
| 0x101 | 1 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |

Zásobník

| | |
|-------|----|
| 0x201 | 0x |
| 0x202 | 1 |
| 0x203 | 0 |
| 0x204 | 1 |
| 0x205 | 0x |
| 0x206 | 1 |
| 0x207 | 0 |
| 0x208 | 2 |
| 0x209 | 1 |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |



pom

Prenos parametrov odkazom - 2

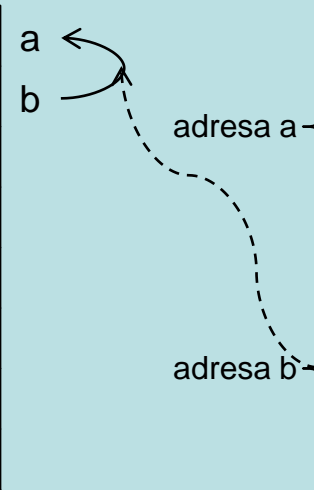
```
void vymena3(byte &a, byte &b)
{
    byte pom = a;
    a = b;
    b = pom;
}

byte a=1, b=2;

int main()
{
    vymena3(a,b);
}
```

Statická paměť

| | |
|-------|---|
| 0x101 | 2 |
| 0x102 | 2 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |



Hodnota z adresy 0x101

Zásobník

| | |
|-------|----|
| 0x201 | 0x |
| 0x202 | 1 |
| 0x203 | 0 |
| 0x204 | 1 |
| 0x205 | 0x |
| 0x206 | 1 |
| 0x207 | 0 |
| 0x208 | 2 |
| 0x209 | 1 |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |

pom

Prenos parametrov odkazom - 3

```
void vymena3(byte &a, byte &b)
{
    byte pom = a;
    a = b;
    b = pom;
}

byte a=1, b=2;

int main()
{
    vymena3(a,b);
}
```

Statická paměť

| | |
|-------|---|
| 0x101 | 2 |
| 0x102 | 1 |
| 0x103 | |
| 0x104 | |
| 0x105 | |
| 0x106 | |
| 0x107 | |
| 0x108 | |
| 0x109 | |
| 0x10a | |
| 0x10b | |
| 0x10c | |
| 0x10d | |
| 0x10e | |
| 0x10f | |
| 0x110 | |
| 0x111 | |

a

b

adresa a

adresa b

Hodnota z adresy 0x101

Zásobník

| | |
|-------|----|
| 0x201 | 0x |
| 0x202 | 1 |
| 0x203 | 0 |
| 0x204 | 1 |
| 0x205 | 0x |
| 0x206 | 1 |
| 0x207 | 0 |
| 0x208 | 2 |
| 0x209 | 1 |
| 0x20a | |
| 0x20b | |
| 0x20c | |
| 0x20d | |
| 0x20e | |
| 0x20f | |
| 0x210 | |
| 0x211 | |

pom

Programátor - praktik

- Myslí za hranice bezprostredného problému
- Posudzuje problém širšom kontexte v rámci celkového obrazu
- Robí rozumné kompromisy a kvalifikované rozhodnutia