

Předávání parametru odkazem

10. cvičení

Jiří Zacpal

KMI/ZP1 – Základy programování 1

2. písemná práce

- Příští hodinu
- Rozdělení na dvě poloviny
- 35 minut na řešení příkladu
- Hodnocení: 0, 3, 5 bodů

Předání parametru hodnotou

- pouze pro **vstupní** parametry funkce
- vhodné pro datové typy s relativně malou velikostí
- formální parametr se definuje jako běžná proměnná

```
int moje_fce(int data) { ... }
```

- skutečný parametr ve volání funkce je libovolný

```
výrazvysl = moje_fce(3*cislo);
```

- hodnota skutečného parametru se při volání funkce ukládá do oblasti parametrů funkce v paměti

Předání parametru odkazem

- lze použít jak pro **vstupní** tak pro **výstupní** parametry
- používá se pro předání pole (nelze předat hodnotou), větších struktur a jakýchkoli typů, mají-li být výstupní
- formální parametr je ukazatel na odpovídající typ

```
int moje_fce(int *data) { ... }
```
- skutečný parametr ve volání funkce je adresní

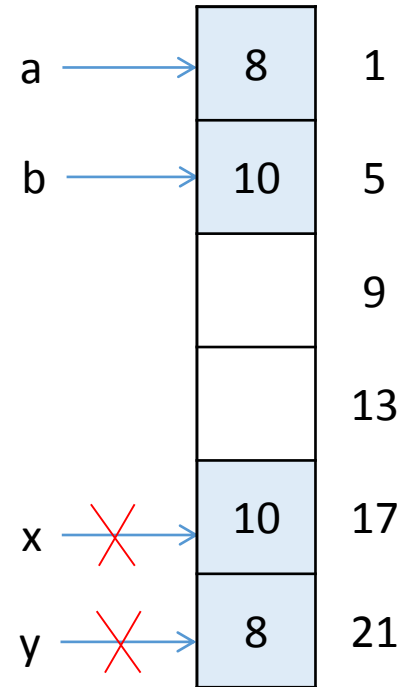
```
výrazvysl = moje_fce(&cislo);
```
- do oblasti parametrů funkce se takto ukládá pouze adresa, data se nekopírují a jejich změny jsou po návratu z funkce zachovány

Příklad 1

Předání parametru hodnotou

```
void vymen(int x, int y)
{
    int pom =x;
    x=y;
    y=pom;
}

main()
{
    int a=8,b=10;
    printf("Puvodni hodnoty:
a=%i, b=%i\n",a,b);
    vymen(a,b);
    printf("Nove hodnoty: a=%i,
b=%i\n",a,b);
}
```

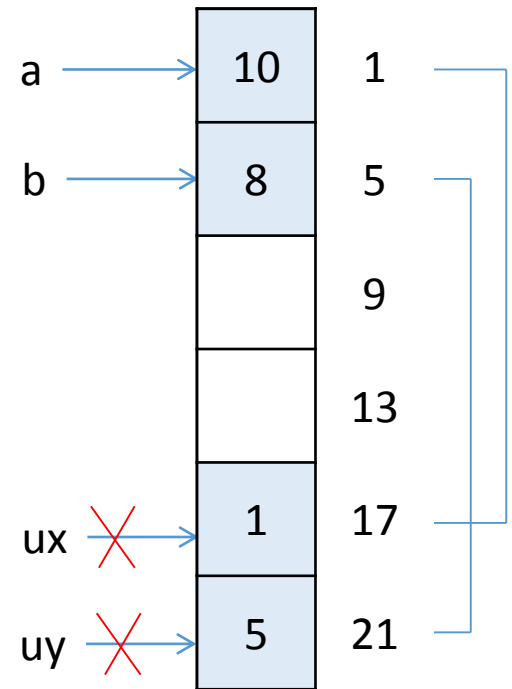


Příklad 2

Předání parametru odkazem

```
void vymen(int *ux, int *uy)
{
    int pom = *ux;
    *ux = *uy;
    *uy = pom;
}

main()
{
    int a = 8, b = 10;
    printf("Puvodni hodnoty: a=%i, b=%i\n", a, b);
    vymen(&a, &b);
    printf("Nove hodnoty: a=%i, b=%i\n", a, b);
}
```



Příklad 3

```
int set(char* in, char** out)
{
    char *r=NULL;
    int i=0,poz=0;
    while(in[i]!='\0')
    {
        if(r==NULL)
            r=(char *)malloc(sizeof(char));
        else
            r=(char *)realloc(r,sizeof(char)*(i+1));
        if(in[i]>='a'&&in[i]<='z')
        {
            r[i]=in[i]-('a'-'A');
            poz++;
        }
        else if (in[i]>='A'&&in[i]<='Z')
        {
            r[i]=in[i]+('a'-'A');
            poz++;
        }
        else r[i]=in[i];
        i++;
    }
    r[i]='\0';
    *out=r;
    return poz;
}

main()
{
    char puv[]="Ahoj svete 23.",*zme;
    set(puv,&zme);
    printf("Puvodni text: %s\n",puv);
    printf("Zmeneny text: %s\n",zme);
}
```