Řetězce

4. února 2012

V minulých přednáškách jsme probírali jednorozměrná a vícerozměrná pole, přičemž jsme se zaměřili pouze na pole typu `int` nebo `double`. Termín řetězec (angl. string) se používá pro speciální pole typu `char`. S řetězci se pracuje trochu odlišným způsobem než s jinými poli a proto je tomuto tématu vyhrazena zvláštní přednáška.

Statické jednorozměrné pole typu `char` můžeme vytvořit stejným způsobem jako pole jiných typů. Například příkaz:

```
char slovo[10];
```

vytvoří pole typu `char` o 10ti položkách. Jednotlivým položkám můžeme přiřazovat hodnotu různých znaků. Přitom je znak vždy nutné uvádět v apostrofech:

```
slovo[0] = 'A';
slovo[1] = 'h';
slovo[2] = 'o';
slovo[3] = 'j';
slovo[4] = '!';
```

Tímto způsobem dostaneme sice jednorozměrné pole znaků, nikoliv však řetězec. V řetězci je zvolená část znaků, kterou chceme používat, ukončena speciálním znakem pro konec řetězce: `\0`. Z pole slovo tedy uděláme řetězec příkazem:

```
slovo[5] = '\0';
```

Výše uvedený způsob definování řetězců není zrovna praktický. Slouží pouze pro ilustraci základního rozdílu mezi jednorozměrným polem typu `char` a řetězcem. Nejjednodušší způsob vytvoření konkrétního řetězce je jeho inicializace zároveň s definicí:

```
char slovo[10] = "Ahoj!";
```

V tomto případě se do pole uloží speciální znak pro ukončení řetězce automaticky. Všimněte si, že na rozdíl od jednotlivých znaků, konstantní řetězec uvádíme v uvozovkách. Při definování velikosti pole slovo je třeba mít na paměti, že jeho velikost musí být minimálně o jednu položku větší než zvolený řetězec, aby bylo místo na speciální znak pro jeho ukončení. Pokud nemáme v plánu vytvořený řetězec dále v programu měnit a tudíž nám stačí velikost odpovídající zvolenému konstatnímu řetězci, můžeme při definici s inicializací velikost pole vynechat:

```
char slovo[] = "Ahoj!";
```

Pozor, již definovaný řetězec můžeme v přiřazovacích příkazech měnit pouze po znacích:

```
slovo[4] = ' ';
slovo[5] = '!';
slovo[6] = '\0';
```

Přiřazení celého řetězce není možné:

```
slovo = "Ahoj !";
```

Snadnou práci s řetězci umožňují funkce `printf()` a `scanf()` a obdobné funkce pro práci se souborem. Pro řetězec je v nich možné použít formát `%s`. Řetězec slovo tedy snadno vypíšeme příkazem:

```
printf( "slovo = %s\n", slovo );
```

Stejně snadno můžeme řetězec načíst z konzole:

```
char slovo[10];
scanf( "%s", slovo );
```

Zde si však všimněte jedné důležité zvláštnosti. V příkazu `scanf()` není před názvem proměnné `slovo` uveden referenční operátor `&`. Je tomu tak proto, že proměnná slovo je v podstatě konstantní ukazatel mající hodnotu adresy, kde je pole uloženo v paměti. Aby funkce `scanf()` mohla do pole zapisovat, je třeba do ní poslat adresu tohoto pole, což je v tomto případě hodnota proměnné `slovo`.

Při načítání z konzole či ze souboru je za řetězec považována řada znaků ukončená jedním z tzv. bílých znaků. Bílým znakem je mezera, tabulátor či odřádkování klávesou enter.