

Argumenty funkce main

8. února 2012

Ačkoliv uživatelé operačního systému Windows jsou zvyklí, že všechny programy mají podobu okénkových aplikací, mnoho programů vyvíjených ve výzkumných ústavech a programů používaných v systému Linux se ovládají z příkazové řádky. Pro zkušené uživatele je totiž jejich ovládání mnohem rychlejší díky používání klávesnice namísto klikání myši. Například na naší katedře mechaniky jsou vyvíjeny dva velké programy pro výpočet konstrukcí metodou konečných prvků: [SIFEL](#) a [OOFEM](#) a oba se ovládají z příkazové řádky. Používání těchto programů je založeno na vytvoření datového vstupního souboru, kde jsou definovány informace o tom, co a jak se má počítat a jaké výstupy se mají kam ukládat. Název vstupního souboru si uživatel může libovolně zvolit. Při spuštění programu z příkazové řádky stačí, když za názvem programu udělá mezeru a poté uvede název vstupního souboru případně dále uvede další parametry vždy oddělené mezerami. Tyto parametry, které jsou uvedeny na příkazové řádce v okamžiku spuštění programu, budou zpracovány. Cílem naší poslední přednášky je naučit se, jak napsat program, který bude používat parametry použité při spuštění z příkazové řádky.

Popsané parametry programu vstupují do hlavní funkce `main` v podobě jejích argumentů. Pokud chceme, aby náš program parametry využíval, musíme funkci `main()` definovat s následujícími argumenty:

```
int main ( int argc, char *argv[] )
```

Tyto argumenty mohou mít libovolný název, ale vždy uvedené datové typy. Přesto se obvykle používají i uvedené názvy. Proměnné `argc` a `argv` jsou lokální proměnné funkce `main`. Po spuštění programu je do proměnné `argc` uložena hodnota odpovídající počtu řetězců uvedených na příkazové řádce v okamžiku spuštění programu. To zahrnuje i samotný název programu. Proměnná `argv` je dvourozměrné pole typu `char`, nebo se na ni můžeme dívat jako na jednorozměrné pole řetězců. Do tohoto pole řetězců jsou všechny řetězce uvedené na příkazové řádce uloženy. Počet řetězců v poli tedy odpovídá hodnotě uložené v proměnné `argc`. Délka jednotlivých řádků pole `argv` odpovídá délce vložených řetězců.

Pro ukázkou zde uvedu program, který vypíše postupně všechny řetězce v poli `argv`.

```
#include<stdio.h>

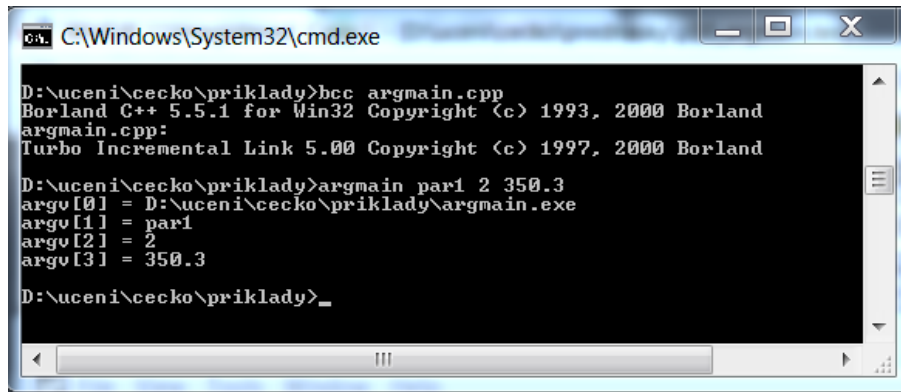
int main ( int argc, char *argv[] )
{
    int i;

    for ( i=0; i<argc; i++ )
        printf( "argv[%d] = %s\n", i, argv[i] );

    return 0;
}
```

Tento zdrojový kód uložíme do souboru `argmain.cpp`, takže po překladači získáme spustitelný soubor s názvem `argmain.exe`. Ten poté spustíme z příkazové řádky se třemi parametry oddělenými mezerami:

```
argmain par1 2 350.3
```



```
C:\Windows\System32\cmd.exe
D:\uceni\cecko\priklady>bcc argmain.cpp
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
argmain.cpp:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland
D:\uceni\cecko\priklady>argmain par1 2 350.3
argv[0] = D:\uceni\cecko\priklady\argmain.exe
argv[1] = par1
argv[2] = 2
argv[3] = 350.3
D:\uceni\cecko\priklady>_
```

Obrázek 1: Spuštění programu.

Výsledek spuštění je ukázán na Obrázku 1.

Na tomto příkladu je ukázáno, že jako parametry programu můžeme uvádět jak znakové řetězce, tak číselné. Pokud dále potřebujeme číselné řetězce převést na odpovídající číselné hodnoty, musíme definovat odpovídající proměnné a tyto hodnoty do nich uložit. K tomu můžeme použít další funkci z knihovny `stdio.h`. Funkce `sscanf()` slouží ke čtení z řetězců. Její použití se podobá použití funkce `fscanf()` pro čtení ze souboru s tím rozdílem, že prvním parametrem není ukazatel na soubor, ale ukazatel na zvolený řetězec:

```
int a;
double b;

sscanf( argv[2], "%d", &a );
sscanf( argv[3], "%lf", &b );
```

Stejně jako v knihovně `stdio.h` existuje funkce pro čtení z řetězců, existuje i funkce pro zápis do řetězců `sprintf()`. Tato funkce nám umožní pohodlné zapisování do řetězců nejen znaků, ale i čísel:

```
int i=1;
char nazev[15];

sprintf( nazev, "Vystup_%03d.dat", i );
```

V tomto případě můžeme vytvořený řetězec uložený v proměnné `nazev` použít třeba pro otevření souboru pro zápis výsledků výpočtu:

```
FILE *f;

f = fopen( nazev, "wt" );
```