

Práce se souborem

14. listopadu 2011

V případě zadávání většího množství údajů, se kterými má program pracovat, je mnohem snazší nechat program načíst potřebné údaje z datového souboru. Naopak v jiných případech je vhodné nechat program větší množství výsledků zapsat do datového souboru. Pro práci se souborem budeme používat funkce, které jsou zahrnuty v knihovně `stdio.h`. Aby mohl program pracovat s datovými soubory, je třeba nejdříve definovat proměnnou, která bude ve zdrojovém kódu příslušný datový soubor reprezentovat. Tato proměnná musí být speciálního datového typu. Následujícím příkazem budeme definovat proměnné `f` a `g` pro reprezentaci dvou různých souborů:

```
FILE *f, *g;
```

Všimněte si, že před názvem každé proměnné musí být vždy uveden symbol `*`. Jeho význam budeme probírat v následující hodině. Dále je nutné připomenout, že jazyk C je citlivý na velikost písmen, takže datový typ `FILE` musí být vždy zapsaný velkými písmeny.

Pro přiřazení daného souboru dané proměnné použijeme funkci `fopen`, která má dva parametry:

```
f = fopen( "název_souboru", "způsob_otevření" );
```

Název souboru musí zahrnovat buď absolutní cestu a nebo relativní cestu k místu, odkud bude program spuštěn. V jazyce C rozlišujeme tři různé způsoby otevření souboru, kterým odpovídá i následná práce s ním: `rt` (z anglického read text) použijeme pro otevření souboru pro čtení, `wt` (z anglického write text) nebo `at` (z anglického add text) použijeme pro zápis. Jednotlivé typy práce se souborem probereme níže. Pro všechny společně platí, že když práci se souborem ukončíme, měli bychom soubor opět zavřít. K tomu slouží příkaz:

```
fclose(f);
```

Čtení ze souboru

Otevření souboru `data.dat`, uloženého ve stejném místě jako program pro čtení, tedy vypadá:

```
f = fopen( "data.dat", "rt" );
```

Pokud má program ze souboru načíst data, která potřebuje k dalším výpočtům, je záhodno včas ověřit, zda se soubor podařilo otevřít. V případě, že se soubor otevřít nepodařilo, funkce `fopen` uloží do proměnné `f` speciální konstantu `NULL`. Pokud další kroky programu nemají bez otevření souboru smysl, můžeme program předčasně ukončit:

```
if ( f==NULL ) {  
    printf( "Soubor se nepodarilo otevřít.\n" );  
    return 0;  
}
```

Když se soubor otevřít podaří, můžeme se pokusit z něj načíst, co potřebujeme. Pro čtení ze souboru se používá funkce `fscanf`, která se velmi podobá funkci `scanf`. Jediný rozdíl spočívá v tom, že prvním parametrem funkce `fscanf` je proměnná, které byl přiřazen soubor, ze kterého se má číst. Příkaz pro načtení reálného čísla do proměnné `a` bude tedy vypadat následovně:

```
double a;
fscanf( f, "%lf", &a );
```

Pokud je v souboru před načítaným číslem jakýkoliv bílý znak (tj. mezera, tabulátor nebo nový řádek), funkce `fscanf` při načítání reálného i celého čísla všechny bílé znaky ignoruje. Pouze načítání znaku do proměnné typu `char` by bílé znaky nepřeskočilo. Jestliže budou v souboru dvě čísla uvedená pod sebou a v programu zavoláme dvakrát za sebou příkaz pro načítání:

```
~
double a,b;
fscanf( f, "%lf", &a );
fscanf( f, "%lf", &b );
```

uloží se první číslo do proměnné `a` a druhé číslo do proměnné `b`. Představme si, jako bychom měli pomyslný kurzor, který určuje, odkud budeme číst. Při otevření souboru je kurzor na začátku souboru a při každém čtení se kurzor posune za část, kterou jsme přečetli. Pokud se při čtení potřebujeme vrátit zpět, nezbývá, než soubor zavřít a otevřít znovu.

Další problém se může objevit v situaci, kdy soubor sice existuje, ale je prázdný. Pokud máme podezření, že by taková situace mohla nastat, můžeme to ověřit tak, že funkci `fscanf` zavoláme jako funkci s návratovou hodnotou, která má typ `int`:

```
double a;
int i;
i = fscanf( f, "%lf", &a );
```

V případě, že se číslo nepodařilo načíst, protože v souboru nic není, funkce `fscanf` uloží do proměnné `i` speciální konstantu `EOF` (zkratka z anglického End Of File). V takovém případě můžeme s použitím podmínky vykonat potřebné kroky, např. opět program předčasně ukončit:

```
if ( i==EOF ) {
    printf( "Soubor je prázdný.\n" );
    return 0;
}
```

Zápis do souboru

Pro zápis do souboru máme dvě možnosti otevření souboru. První je otevření souboru v módu `wt`:

```
f = fopen( "data.dat", "wt" );
```

Pokud v tomto případě soubor neexistuje, vytvoří se nový. Pokud takový soubor již existuje, přepíše se, takže o data, která v něm byla uložena, přijdete. Proto je důležité si na tento typ otvírání souboru dávat pozor.

Druhou možností je otevření souboru v módu `at`:

```
f = fopen( "data.dat", "at" );
```

Pokud v tomto případě soubor neexistuje, opět se vytvoří nový. Pokud však existuje, uložena data v něm zůstanou a při zápisu se další údaje začnou zapisovat na jeho konec.

Pro zápis do souboru existuje funkce `fprintf`, která je opět velmi podobná funkci `printf` s tím rozdílem, že jejím prvním parametrem je proměnná přiřazená zvolenému souboru. Příkaz, který do souboru zapíše hodnotu reálné proměnné `a` vypadá tedy takto:

```
fprintf( f, "%g", a );
```