

Prostředí, ve kterém budeme pracovat

7. listopadu 2011

V současné době existuje mnoho různých prostředí vhodných pro programování v jazyce C++ a většinou se liší pro různé operační systémy. Tato vývojová prostředí mají dvě základní funkce:

- editor pro psaní zdrojového kódu programu
- překladač pro překlad zdrojového kódu programu do spustitelné verze programu.

Kromě těchto dvou základních funkcí poskytují prostředí často i další, ale ty nejsou pro výuku základů programování nezbytné a proto je v rámci tohoto kurzu nebudeme využívat.

Editor

Aby výuka zůstala v tomto ohledu pokud možno co nejvíce univerzální, budeme používat editory, které jsou volně šiřitelné a dostupné pro operační systém Windows i Linux. Jedná se v podstatě o jednoduché textové editory, které podporují zvýrazňování klíčových slov programovacího jazyka C++ a také inteligentní zarovnávání zdrojového kódu, které začátečníkům velmi pomáhá. Na stránkách předmětu jsou umístěny instalace dvou níže popsanych editorů.

jEdit

jEdit je jednoduchý editor podporující většinu standardních klávesových zkratk, např. pro kopírování, vkládání atd. jEdit je vytvořen v programovacím jazyce Java a proto vás možná vyzve k doinstalování některých nástrojů pro tento jazyk.

Aby správně fungovalo zvýrazňování klíčových slov, je nutné, abyste váš zdrojový kód vždy uložili do souboru s koncovkou .c nebo .cpp. Pokud na libovolném řádku zdrojového kódu stisknete zkratku `ctrl+i`, příkaz na tomto řádku se vám automaticky zarovná do správné pozice z hlediska štabní kultury pro programovací jazyk. Tato funkce je velmi vhodná pro odhalení základních syntaktických chyb ve vašem kódu, neboť pokud někde zapomenete středník či závorku, příkaz se zarovná jinak a vy můžete snadno a rychle odhalit chybu. Protože byste tuto výhodu měli v počátcích využívat často, je klávesová zkratka `ctrl+i` trochu nevhodná. Proto si ji můžete nastavit v nabídce Utilities->Global Options, v záložce Shortcuts. Jedná se o příkaz Indent Lines. Poklikáním ji můžete změnit. Doporučuji nastavit na klávesu TAB.

Další vhodnou úpravou je nastavit velikost odsazení. Implicitně je nastaveno odsazení 8 znaků, což vede k tomu, že každý for cyklus vás bude odsouvat velmi daleko a při použití tří vnořených for cyklů bude délka řádku vašeho programu již nepříjemně dlouhá. Toto je možné změnit opět v nabídce Utilities->Global Options, v záložce Editing. Stačí změnit hodnoty Tab width a Indent width na menší hodnotu, doporučuji 2.

XEmacs

XEmacs je trochu náročnější editor a v počátcích jsou jeho velkou nevýhodou rozdílné klávesové zkratky pro editaci (kopírování, vkládání atd.) od běžně používaných v systému Windows. Na druhou stranu podporuje tento editor mnoho různých funkcí a kromě podpory programovacích jazyků podporuje také

typografický nástroj \LaTeX , který je vhodný pro tvorbu odborných textů, zejména pokud obsahují mnoho matematických výrazů.

I pro tento editor platí, že podbarvení klíčových slov funguje až v okamžiku, kdy je soubor uložen s patřičnou koncovkou .c nebo .cpp. Dále může být ještě potřeba v nabídce Options->Syntax Highlighting zaškrtnout položku In This Buffer. V nabídce Options je možné nastavit mnoho dalších věcí, ale při dalším spuštění programu se automaticky obnoví výchozí nastavení. Naše nastavení je ale možné uložit tak, že v nabídce Options spustíte položku Save Options to Custom File. Aby toto ovšem fungovalo v systému Windows, je nejdříve nutné vytvořit ve vaší uživatelské složce `C:\Users\xxx\` adresář `.xemacs`, kde `xxx` je vaše uživatelské jméno.

Protože překladač vám bude vždy uvádět číslo řádku, na kterém máte nějakou chybu, je dobré si nastavit zobrazení čísel řádků v nabídce Options->Display->Line Numbers.

XEmacs také podporuje inteligentní zarovnávání jednotlivých řádků zdrojového kódu. Bez ohledu na počet mezer, které napíšete, jakmile na daném řádku stisknete klávesu TAB, řádek se zarovná na pozici, kam logicky patří. Toho v začátcích často využívejte, abyste tak snadno odhalili chybějící závorku či středník, který způsobí, že se následující řádek zarovná na špatnou pozici.

Překladač

Kód programu v jazyce C nebo C++ je třeba přeložit do spustitelné verze programu pomocí nějakého překladače. Překladačů je mnoho a v různé míře respektují normu ANSI/ISO. Pro výukové účely jsem pro studenty pracující v systému Windows vybrala překladač Borland C++ compiler 5.5. Tento překladač je volně dostupný a můžete si ho stáhnout ze stránek předmětu. Jeho instalace má bohužel pár drobných nedostatků, které jsem se částečně pokusila odstranit doplňkovým programkem `bcc.exe`, který je rovněž na stránkách předmětu. Aby vše dobře fungovalo, je třeba překladač nainstalovat do cesty, kterou sám nabídne, tj. přímo na systémový disk (dále budu předpokládat, že systém máte na disku C). Po instalaci překladače uložte program `bcc.exe` do adresáře `C:\Borland\BCC55\Bin\`.

Samotný překlad budeme provádět na příkazovém řádku, který otevřete např. tak, že v nabídce Start máte zjednodušenou příkazovou řádku, do které napíšete příkaz `cmd` a tím se vám příkazová řádka otevře. Na příkazové řádce budete implicitně ve svém uživatelském adresáři ve složce Users. Pokud si svoje zdrojové kódy budete ukládat někam jinam, musíte do příslušné složky přejít. Příkazem `cd ..` můžete vyskočit z aktuální složky o úroveň výš. Samotným příkazem `d:` přejedete na disk D. Příkazem `cd xxx` otevřete složku `xxx`, pokud ta existuje atd. Pokud používáte Windows Commander nebo Total Commander, tak tyto aplikace mají v dolní části jednoduchou příkazovou řádku. Pokud v ní spustíte příkaz `cmd`, pak se s příkazovou řádkou ocitnete právě v adresáři, ve kterém jste právě byli v Commanderu. Můžete si tak usnadnit pracné hledání adresáře s vašimi zdrojovými kódy pomocí příkazů `cd`.

V této chvíli, když na příkazové řádce spustíte překladač příkazem `bcc`, dostane se vám odpovědi `'bcc' is not recognized as an internal or external command...` nebo nějaké podobné říkající, že příkaz je neznámý. Aby bylo možné ve vaší složce překladač spustit, je ještě nutné přidat jeho cestu do systémové proměnné Path. To provedete následujícím způsobem. V nabídce Start otevřete Ovládací panely->Systém->Upřesnit nastavení systému->Proměnné prostředí. V dolní části okna budete mít seznam systémových proměnných, ve kterých najdete proměnnou Path a klikněte na Upravit. Všimněte si, že v hodnotě proměnné jsou uvedeny cesty ke složkám, kde jsou uloženy spustitelné soubory všech programů, které jste nainstalovali. Jednotlivé cesty jsou oddělené středníkem bez mezery. Proto stejným způsobem připište na konec celého řetězce cestu k překladači, takže na konci budete mít `;C:\Borland\BCC55\Bin`. Pokud máte v tuto chvíli otevřenou příkazovou řádku, tato změna se v ní neprojeví a proto je nutné ji nejdříve zavřít a otevřít znovu. Nyní zkuste, zda se vše podařilo tak, že opět spustíte příkaz `bcc`. Pokud vše proběhlo správně, měla by se vám objevit odpověď:

```
Error: No input file given.
```

```
Usage: bcc file.c.
```

Překladač vám tak oznamuje, že jste nezadali název žádného zdrojového kódu, který by se měl přeložit a že pro překlad máte za příkazem `bcc` uvést název zdrojového kódu. Pokud bude překlad úspěšný, vznikne kromě pár dalších souborů spustitelný soubor `file.exe`, který spustíte z příkazové řádky příkazem `file`.

Koncovku `.exe` není při spouštění nutné uvádět.

V systému Linux je překladač většinou standardní součástí instalace. V tomto případě budeme používat překladač `g++`. Celý příkaz pro překlad vypadá následovně:

```
g++ -o file file.c
```

kde `file.c` je název zdrojového kódu a `file` je název spustitelného souboru. V Linuxu spustitelné soubory nemají koncovku `exe`, většinou nemají koncovku žádnou. Soubor je možné spustit pomocí jen jeho jména nebo bude možná třeba před jeho jméno napsat ještě tečku a lomítko:

```
./file
```

V tuto chvíli máme všechno nachystané a můžeme se pustit do programování.