Автор: Андрей
28.10.2009 09:27 - Обновлено 02.03.2010 11:25

Истоки проблемы

Любите ли вы русские символы в консольных приложениях так, как люблю их я? Всегда приятно, когда программа общается с тобой не на языке международного общения, коим является английский язык, а на родном и понятном русском! Но вот незадача: создаваемые на просторах Соединенных Штатов Америки среды разработки прекрасно поддерживают английский язык, русский же такого фавора у них не имеет.

Какие же именно неприятности нас могут подстерегать и зачастую подстерегают? Чтобы понять это, попробуйте в консольном приложении в C++ Builder'е вывести любую строку на русском языке. В зависимости от ситуации программа выведен вам тарабарщину, абракадабру или кракозябры. В "родных" для Билдера приложениях с формами-кнопками и тому подобным этой проблемы не наблюдается. В чем же дело?

А дело на самом деле в том, что C++ Builder использует Windows-кодировку символов (ANSI), а консольные приложения - DOS-кодировку (ASCII). Т.е. если при написании кода в редакторе Билдера вы где-то определили строку с русскими символами, то последовательность битов, в которой будут храниться эти символы, будет определяться кодировкой ANSI, однако консольное приложение будет использовать при их интерпретации кодировку ASCII, свято веря, что вы именно этого от него и хотите. Символам латиницы от этого недопонимания ни жарко ни холодно - их коды в обеих кодировках совпадают. Русские же символы ощущают на себе все трудности перевода...

Разглагольствования об альтернативах

Итак, что же делать? Писать транслитом? Как полумера - годится, однако же лично меня такая полумера не устраивает и навевает на меня тоску. Где ты, о великий, могучий, правдивый и свободный русский язык? Неужели же бездушная машина навяжет мне бесконечное над тобой надругательство, заведет меня в "neprolaznuyu chaschu", из которой не выбраться никогда? Нет, транслит - не вариант.

Пользователи MS Visual Studio утверждают, что заставить программу говорить на русском, а не на неизвестном науке языке можно с помощью строчки

```
setlocale(LC_ALL, "Russian");
```

Не знаю, с Visual Studio не встречался. Но в моем родном C++ Builder 6 это точно не

работает.

Итак, насколько мне известно, Borland C++ Builder не предоставляет средств, способных непосредственно решать данную проблему. Поэтому нам остается надеяться только на себя, да на силу русского оружия (главнейшие из которых - острый ум, изобретательность и смекалка - а всякие там единороги Шувалова, Тетрисы, да Тополь-М - лишь производные).

Предварительная подготовка

Итак, пойдем от печки. Что нам нужно, чтобы иметь возможность использовать русские символы при наличии несовпадающих кодировок? Очевидно, можно написать функцию перевода из одной кодировки в другую, которую впоследствии можно либо добавлять в каждую нуждающуюся в ней программу, либо поместить в отдельный модуль и уже его по мере необходимости подключать к различным программам.

Однако чтобы переводить символы из одной кодировки в другую, нам нужно как-то установить между ними соответствие. Здесь вам могут помочь таблицы символов для каждой из этих кодировок, однако я опишу способ гораздо более простой и не требующий никакой технической информации.

Прежде всего нам потребуются два массива символов - русский алфавит в одной кодировке, и в другой. Для этого прежде всего создадим текстовый документ (*.txt) и откроем его с помощью WordPad. Запишем в документ две строки

"абвгдеёжзийклмнопрстуфхцчшщъыьэюя" "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭ

или одну

"абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮ

что впоследствии будет совершенно без разницы. Сохраним этот файл в формате ASCII (тип файла - "текстовый документ MS-DOS" в настройках сохранения). Теперь у нас есть русский алфавит в ASCII. Создадим теперь в C++ Builder консольное приложение, которое будет использовать функцию и сохраним его где-нибудь, после чего закроем. Теперь скопируем наш текстовый документ в папку только что созданного проекта и переименуем этот текстовый документ в (по умолчанию) Unit1.cpp, заменив тем самым изначальный файл. Вновь открываем проект и - вуаля! - у нас есть ASCII-русский алфавит непосредственно в нашей программе (не смущайтесь этих кракозябр: эти - хорошие!). Зачем переименовывать файл и заменять файл исходника, почему бы просто не скопировать из текстового документа в текст программы? Попробуйте, и посмотрите что получится!

Функция перевода из одной кодировки в другую

Итак, теперь ничего не стоит написать функцию, которая переводила бы русские символы из одной кодировки в другую. Надо только оформить наши кракозябры

константным массивом символов, добавить такой же массив вполне читабельных ANSI-буковок и функцию, которая будет все это использовать. Получится примерно следующее:

```
        const int N = 66;
  const char DosABC[N] =
  "" //Здесь были хорошие кракозябры, но, к сожалению, при копировании они
  ""; //испортились и я их удалил. Вы можете сделать их сами, как - см. выше
  const char WinABC[N] =
  "абвгдеёжзийклмнопрстуфхцчшщъыьэюя"
  "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

    std::string ToDosStr(std::string input)
     {
     std::string output = "";
     bool Ok;
     for (unsigned i = 0; i          {
        Ok = false;
        for (int j = 0; j                    if (input[i] == WinABC[j])
              {
              output += DosABC[j];
              Ok = true;
              }
        if (!Ok)
            output += input[i];
        }
     return output;
     }
```

Теперь ничего вам не мешает написать

```
       std::cout  pass;
  if (ToWinStr(pass) == "ракета")
    std::cout
```