Структура файлов сообщений

2004-10-07 19:05:18

Yuri Prokushev

В данной статье приведено описания строения файлов *.MSG, библиотеки MSG.DLL. Дано описание компилятора MKMSGF.EXE и формата исходных файлов сообщений.

Внутренности NLS

В рамках проекта osFree стоит задача разработки полнофункциональной замены стандартных утилит разработки. Одной из таких утилит является аналог компилятора файлов сообщений с открытым исходным кодом. В результате данной работы было проведено исследование структуры файлов *.MSG. Подготовка исходных файлов сообщений

Исходным файлом для компилятора файлов сообщений является обычный текстовой файл. Формат файла довольно простой.

; Коментарий начинается с символа ';'. Строка с данным символом пропускается. MSG MSG0001I: Информационное сообщение %1. MSG0002E: Сообщение об ошибке %1 %2. MSG0003H: Большое многострочное сообщение MSG0004P: Сообщение без перехода на новую строку %0

Здесь MSG - это идентификатор файла сообщений. Идентификатор должен состоять из 3-х символов латинского алфавита. Оригинальный компилятор не позволяет вставлять коментарий между идентификатором файла сообщений и идентификатором сообщения. Аналог данного компилятора позволяет использовать коментарий в любой части файла.

MSG0001I - составной идентификатор сообщения. Первые три символа - идентификатор файла сообщений. Следующие четыре символа - номер сообщения. Последний символ - тип сообщения. Может быть любым латинским символом из

```
I    Информационное сообщение
H    Справочное сообщение
E    Сообщение об ошибке
W    Предупреждение
P    Prompt
?  Нет сообщения
```

Идентификатор сообщения заканчивается символами двоеточия и пробела. Для вставки параметров в сообщения предусмотрены модификаторы %?, где ? равен числу от 1 до 9. Специальный символ %0 используется для типа сообщения Prompt, для предотвращения вывода символа перевода строки. Сообщение Prompt в обязательном порядке содержит в конце последовательность %0.

Компиляция файла сообщений

Для компиляции файла сообщений необходима утилита MKMSGF из OS/2 Developer's Toolkit. Так как Toolkit не распространяется свободно (исключая владельцев eComStation), была написана аналогичная по своим возможностям утилита под osFree License. Использование обеих утилит полностью одинаково.

Предположим, что файл с исходными сообщениями называется NOS_RU.txt, а двоичный файл сообщений имеет имя NOS_RU_RU.MSG. Тогда для компиляции необходимо использовать команду

```
MKMSGF NOS_RU.txt NOS_RU_RU.MSG
```

При этом файл будет создан с кодом текущей страны. Для создания прочих языковых файлов необходимо использовать дополнительные ключи:

```
/V   Вывод дополнительной информации
/D   подмножество  DBCS  или код страны с  DBCS
/P   Кодовая страница. Может быть указано до 16 страниц
/L   Идентификатор языка и диалекта
/?  Вывод справочной информации
```

Многоязычные файлы сообщений

Файлы сообщений не были бы так интересны, если бы могли содержать только сообщения одной страны, в одной кодировке. Поэтому существует возможность создания многоязыковых файлов сообщений. При этом подготавливаются аналогичные исходные файлы сообщений с одинаковыми идентификаторами сообщений, но с различным текстом сообщений. Прежде чем начинать компиляцию файлов сообщений, необходимо подготовить управляющий файл. Например, файл NOS_UNI.txt:

NOS_EN.txt NOS_UNI.MSG /L1 /P437 NOS_RU.txt NOS_RU_RU.cp866.MSG /L7 /P866 NOS_RU.koi8r.txt NOS_RU_RU.koi8r.MSG /L7 /P878 NOS_RU.win1251.txt NOS_RU_RU.win1251.MSG /L7 /P1251

После запуска компилятора

MKMSGF @NOS_UNI.txt

создаются файлы сообщений NOS_UNI.MSG и NOS_RU_RU.*.MSG. Файл NOS_UNI.MSG будет являться опорным файлом и содержать информацию об остальных файлах сообщений, то есть в дальнейшем не потребуется перебирать все варианты файлов для поиска необходимого.

Встраивание файлов сообщений в исполняемый файл

Возможно встраивание файла сообщений в исполняемый файл. Для этого предназначена утилита MSGBIND.EXE. Данная статься не рассматривает данный процесс, отметим только, что сообщения сохраняются в сегменте MSGSEG.

Использование файлов сообщений

Для управления файлами сообщений используется библиотека MSG.DLL. Данная библиотека содержит четыре функции:

```
DosTrueGetMessage
DosInsertMessage
DosPutMessage
DosIQueryMessageCP
```

Функция DosTrueGetMessage обычно напрямую не используется. Различные компиляторы обычно оспользуют механизмы runtime-библиотек для предоставления функции DosGetMessage, которая не содержит параметр MsgSeg, указывающий на сегмент с сообщениями, встроенными в исполняемый файл.

Аналогичный подход, по тем же самым причинам, выполняется для функции DosQueryMessageCP, которая скрывает вызов функции DosIQueryMessageCP.

Функции DosInsertMessage и DosPutMessage используются напрямую без необходимости какого-либо вмешательства со стороны runtime-библиотек.

Более подробное описание данных функций и пример реализации runtime-части для конкретного компилятора может быть получено из Developer's Toolkit и исходных текстов runtime для OpenWatcom, EMX, VirtualPascal, FreePascal, Sibyl и ряда других компилятороа.
Формат двоичного файла сообщений

Заголовок двоичных индексированных файлов сообщений OS/2 имеет следующую структуру:

```
Magic               : Array[1..8] of Char; // Сигнатура файла
Identifier          : Array[1..3] of Char; // Идентификатор сообщений
                                           //       (SYS, DOS, NET и пр.)
MessagesNumber      : Word;                // Количество сообщений
FirstMessageNumber  : Word;                // Номер первого сообщений
Offsets16bit        : Boolean;             // Размерность таблицы смещений
Version             : Word;                // Версия файла 2 - Новая 0 - Старая
IndexTableOffset    : Word;                // Смещение таблицы индекирования
CountryInfo         : Word;                // Смещение информации о стране
NextCountryInfo     : DWord;               // Смещение списка информации
                    // для многоязычный файлов сообщений
Reserved2           : Array[1..5] of byte; // Назначение неизвестно
```

Файлы старой версии сейчас практически не используются. Единственное приложение, использующее формат старой версии (OS/2 1.x) - это Software Installer. Все поля, начиная с Version, для версии 0 заполняются нулями.

Для многоязычных файлов сообщений поле NextCountryInfo содержит смещение на список файлов сообщений и информацию о стране для этих файлов. Данный список содержится в одном, главном, файле сообщений. Остальные файлы данного списка не содержат. Список начинается со следующей структуры

```
BlockSize         : Word; //Размер блока информации о стране
BlocksCount       : Word; //Количество блоков
```

Блок информации о стране имеет следующую структуру:

```
BytesPerChar      : Byte;                  // Байт на символ (1 - SBCS, 2 -
DBCS)
Reserved          : Array[0..1] of byte;   // Неизвестно
LanguageFamilyID  : Word;                  // Семейство языков
LanguageVersionID : Word;                  // Диалект языка
CodePagesNumber   : Word;                  // Число кодовых страниц
CodePages         : Array[1..16] of Word;  // Список кодовых страниц (макс. 16)
Filename          : Array[0..260] of Char; // Имя файла сообщений
```

Новый формат исходного файла сообщения

Вы могли заметить, что исходный формат файла не очень-то и удобен. Синхронизировать файлы на различных языках довольно сложно. Известные автору прочие подходы к исходным файлам (например, tmf, gettext) либо вообще не поддерживают отличную от 850 кодовую страницу, либо поддерживают всего одину кодовую страницу на файл. Поэтому предложен свой формат, который использует кодировку UTF-8, а следовательно, поддерживает практически неограниченное число кодовых страниц. Кроме того, поддержка национальных файлов значительно упрощается за счет совмещения всех языков в одном файле.

Поленившись изобретать велосипед, автором предложено использовать язык xml для разметки файла сообщения, тем более, что в наличии имеются различные библиотеки для работы с xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- osFree message file
-->
<!--
-->
<!-- Messages from 0000 to 7999 are reserved for compatability reason with
-->
<!-- future versions of OS/2 and eComStation. If you want add new messages,
-->
<!-- then add them starting to 8000-9999 range.
-->

<messagefile id="SYS">
  <languages>
    <language id="en" codepage="850" countrycode="1" countrysubcode="1"/>
    <language id="ru" codepage="866" countrycode="7" countrysubcode="1"/>
  </languages>
  <messages>
    <msg number="0" type="I">
      <lang id="en">Y N A R I<br /></lang>
      <lang id="ru">1 2 1 2 3<br /></lang>
    </msg>
```

```
    <msg number="1" type="I">
      <lang id="en">Incorrect function<br /></lang>
      <lang id="ru">Неверная функция<br /></lang>
    </msg>
  </messages>
</messagefile>
```

Как вы можете видеть, здесь два логических блока:

```
Блок определения языков
Блок описания сообщений
```

Блок определения языка аналогичен файлу ответов для многоязычных файлов сообщений.

Расширение API

Наверное, единственное, что хотелось бы расширить - это ввести аналогичную GNU gettext функцию. Т.е. получать сообщение не только по индексу, но и по сообщению первого языка. Предлагаемый прототип функции:

```
APIRET APIENTRY  DosGetMessage(PCHAR* pTable,
                               ULONG cTable,
                               PCHAR pBuf,
                               ULONG cbBuf,
                               PSZ   pszMessage,
                               PSZ   pszFile,
                               PULONG pcbMsg);
```

Таким образом, при вызове функции DosGetMessage(nil, 0, buf, sizeof(buf), “English message\n”, “OSO001.MSG”, nil) будет получено, например “Русское сообщение\n”. Недостатком такой функции является более низкая скорость работы, но улучшение читаемости кода, а также возможность автоматического получения строк из исходного кода.

Пришлите ваши комментарии

Автору интересны предложения и замечания по новому формату файлов сообщений, равно как и предложения по расширению существующего API.

Читайте также

```
Локализация с помощью GNU GETTEXT
Говорите по-русски
 Multilingual Resources
Работа с кодовыми страницами стандартными средствами OS/2
```