Архитектура ОС UNIX

Санкт-Петербургский государственный политехнический университет

8 декабря 2011

Локализация

Термин интернационализация (i18n) — способ проектирования (дизайн) ПО, при котором возможность многоязыковой поддержки закладывается с самого начала. Локализация (localization, или сокращенно l10n) — это процесс адаптации ПО под конкретные национальные требования . А технически, локализация — это изготовление отдельных объектов локализации в соответствии с требованиями конкретного языка.

POSIX locale

В стандарте POSIX (Portable Operating System) были определены средства локализации, названные locale.

POSIX locale

Структурно, средства POSIX locale состоят из следующих компонентов :

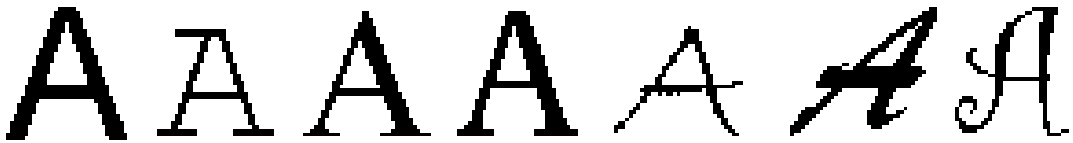
- Набор библиотечных (libc) вызовов (locale API): setlocale(), isalpha(), toupper().
- Переменные окружения, для управления средствами locale : LANG, LC_CTYPE.
- Утилита для получения информации о средствах locale : locale
- Исходные тексты описания locale : locale definitions, Character Set Definition File
- Утилита для изготовления (компиляции) объектов локализации : localedef
- Объекты локализации (наборы данных locale) : /usr/share/locale/

Языки, символы и кодировки

Информация, полученная из «человеческого» мира и предназначенная для машинной обработки, как правило имеет специальный арибут : язык или language (lang).
Большинство языков мира имеют письменнось, а некоторые языки даже несколько, системы письменности существуют самые разнообразные, но большинство письменностей — это изображение последовательности специальных символов (character).

Языки, символы и кодировки

Каждый «абстрактный» символ имеет изображение — glyph.
Считается, что каждый символ имеет «каноническое» изображение , то есть такое, которое позволяет однозначно идентифицировать данный символ.
Таким образом, в модели POSIX и UNICODE не уделяется никакого внимания вариантам начертания символа, то есть шрифтам (fonts) во всем их многообразии.



Языки, символы и кодировки

В стандарте UNICODE кроме определенного изображения каждому символу присвоено определенное имя, например, A — U+0041 — LATIN CAPITAL LETTER A
Таким образом, для символ (character) — это единица тектовой информации, имеющая определенное изображение и определенное имя.

Языки, символы и кодировки

Наиболее важным понятием при обработке символов является понятие Coded Character Set (CCS).
Каждому символу соответствует определенный код, который должен быть сопоставлен с его символом, при поможи таблицы соответствия.
Набор символов, кодов и их соответствий называется charset.

В стандартном POSIX (например на stdin/stdout) мы имеем только потоки кодов. А вся информация о CCS потеряна. Подробнее про это можно прочитать в статье о локализации POSIX.

Языки, символы и кодировки

Существует минимальный (переносимый) (POSIX) Portable Charset - набор символов, который должны поддерживать любые информационные системы (определен в стандарте ISO 646) (он же ASCII).

Для установки в POSIX-системе новой locale с другим набором символов (charset) применяется утилита localedef для компиляции Файла Описания Локализации и Файла Описания Набора Символов (Character Set Description File).

```
localedef -c -i ru\_RU -f KOI8-R ru\_RU.KOI8-R
```

Для проверки установленных в POSIX-системе charset-ов применяется утилита locale:

```
locale -m
```

Категории локализации

Каждой отдельной стране способ представления данных отличается. Чтобы упорядочить все эти отличия, их объединяют в группы и особым образом описывают.

- LC_CTYPE — определяет правила классификации и преобразования одиночных символов. Включает правильную работу isalnum(), isalpha(), iscntrl(), isdigit() и правильный перевод строчных — прописных букв: toupper() и tolower().
- LC_COLLATE — определяет правила сравнения и преобразования строк. Позволяет определять порядок сортировки в местном алфавите, включает правильную работу strcoll() и strxfrm().

Категории локализации

Каждой отдельной стране способ представления данных отличается. Чтобы упорядочить все эти отличия, их объединяют в группы и особым образом описывают.

- LC_NUMERIC — определяет правила национального представления чисел с плавающей точкой. Влияет на strtod() и форматы %f и %g printf(), scanf().
- LC_MONETARY — определяет правила национального представления денежных величин, strfmon().

Особая категория локализации LC_ALL служит для обращения одновременно ко всем категориям

NLS

NLS (National (или Native) Language Support) — возможность получения сообщений и работы с программой на родном языке. «Национализация» в действительности включает в себя несколько аспектов:

- Перевод сообщений программы на национальный язык.
- Присвоение значения категории локализации LC_MESSAGES и/или переменной LANG.
- Перевод описания программы для man.

Чтобы организовать работу с сообщениями программы и дать возможность переводить их на другие языки, было введено такое понятие, как message catalog (каталог сообщений). Message catalog — это база данных, выборка из которой происходит по ключам:

- выбранный пользователем язык сообщений;
- имя исполняемой программы;
- конкретное сообщение в программе.

В результате, на выходе, мы получаем строку конкретного сообщения конкретной программы на нужном нам языке.

NLS

Система NLS предоставляет набор стандартных библиотечных вызовов для работы с каталогами сообщений и набор утилит для создания и поддержания этой базы.
В настоящее время существут две основных реализации NLS:

- X/Open XPG3/XPG4 (с функциями catopen(),catgets(),catclose() и утилитой gencat).
- SUN XView (с функциями gettext(), textdomain()) .

Для поиска самих message catalog-ов от данной программы используется переменная окружения NLSPATH.

Как включить локализацию

Если на UNIX машине средства locale правильно установлены и программы правильно написаны, то локализация включается путем задания строки окружения LANG:

---

export LANG={язык}

Если такой строки окружения нет, работает значение локализации по умолчанию : LANG="C"или LANG="POSIX".

Как включить локализацию

По стандарту POSIX значения локализации записываются в форме: language_TERRITORY.Codeset
Стандарта на названия Codeset-ов формально нет, но принято пользоваться названиями, зарегистрированными в IANA или в RFC-1700.
Для русского языка переменная LANG как правило устанавливается: ru_RU.utf8, ru_RU.KOI8-R

Написание программ с поддержкой интернациализации

«Правильно» написанная программа с использованием POSIX locale не должна зависить от способа кодирования символов. Такая программа нигде не привязана к 7-ми битности ASCII символов, и пользуется стандатрными библиотечными (API) функциями locale.

Не правильно:

```
if (c >= 'A' && c <= 'Z') { ...
```

Правильно:

```
if (isalpha(c) && isupper(c)) { ... или
if (isascii(c) && isupper(c))
```

Написание программ с поддержкой интернациализации

Программа должна явно начинаться с вызова setlocale(LC_ALL,) (такая форма вызова означает, что всем категориям локализации одновременно будет присвоено значение переменной окружения LANG).
Для получения locale-зависимой информации следует пользоватся данными структуры lconv, которые можно получить вызовом функции localeconv() . Для получения детальной информации по категориям локализации (описанным в файле <langinfo.h>) можно пользовться функцией nl_langinfo().