BSD/I18N — Internationalization of the 4.3BSD UNIX System

Zhou Jianqiang (周建强）
Dept. of Computer Science, Nanjing University, Nanjing 210008
Yang Liqun (杨立群) and Pan Shilei (潘世雷)
Zhongji (Omron Shanghai) Computer Company, Shanghai 200030
Tan Hong (谈宏)
Dept. of Computer Science, Shanghai Jiaotong University, Shanghai 200030
Received March 10, 1992; revised May 7, 1992.

Abstract

BSD/I18N project is established for the purpose of internationalizing the 4.3BSD UNIX systems. Its tasks are to provide an internationalized application software development environment, support the 4.3BSD commands/utilities with multi-language processing, and build a complete English and Japanese user environments without excluding other languages like Chinese and Korean.

Keywords: Internationalization, UNIX system, character set, environment.

1. Introduction

In the past, UNIX or UNIX-like systems have been established based on the 7-bit ASCII coded character set and American-English. Much work has been done to transport these systems to other local environments individually, where languages and cultural conventions are different, and additional or alternative character sets are required.

Internationalization aims at supporting programs with multi-language operation in order to achieve international portability of application softwares. It can be done by writing a program which makes no hard-coded assumptions about language, cultural conventions or character encodings. Such a program is said to be internationalized. Data specific to each supported language, territory and codeset combination are held separate from the program code and can be bound to its runtime environment by the language initialization functions.

BSD/I18N project is established for the purpose of internationalizing the 4.3BSD UNIX systems. Its tasks are to provide an internationalized application software development environment, support the 4.3BSD commands/utilities with multi-language processing, and build a complete English and Japanese user environment without excluding other languages like Chinese and Korean. This paper presents our design, implementation, and verification processes of the project. It is organized as follows. Section 2 gives the objectives and tasks of the project. Sections 3 and 4 detail its design, implementation, and verification. Discussions are found in the last section.

2. BSD/I18N Project

2.1. Project Tasks

(1) Developing Localization Facilities

Localdef and GencaT should be provided for a user to define his localization data, that is, the data specific to collating sequence, character classification and case conversion information, language information, and message translations. It is this information that is loaded into a program’s Locale by the language initialization functions and accessed by the program at its runtime to determine its locale-sensitive behavior.

Consideration is also offered to the current situation of the Japanese environment where
several codesets, like ujis, jis7, are widely used. Therefore, Iconv and Iconvdef, the general codeset conversion facility, will be supported by the project.

2. Developing Library Functions

Following classes of functions should be supported by the project to provide internationalized application software development environment (See Table 1).

(a) Functions to initialize a program's locale.
(b) Functions to provide character classification, case conversion, and number format conversion.
(c) Functions to support string comparison based on collation.
(d) Functions to access cultural and language specific data.
(e) Functions to support wide studio operations on streams.
(f) Functions to support multibyte characters.
(g) Functions to support wide character string operations.
(h) Functions to support general codeset conversions.
(i) Functions to support message catalogues.
(j) Functions to support regular expression matching.
(k) Functions to provide file name expansions.

3. Internationalizing the 4.3BSD Commands/Utilities

Most of the locale-sensitive commands/utilities in the 4.3BSD systems should be internationalized according to each locale category they depend on. They are classified into five classes, representing different styles of internationalization and different extents of complexity to modify the existent source code. Table 2 shows all of the 4.3BSD commands/utilities to be internationalized and the categories they depend on. Since all of the commands/utilities are Lc-messages (message catalogue) dependent, only those dependent on yes/no expression message catalogue are listed in Table 2.

<table>
<thead>
<tr>
<th>No</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>setlocale</td>
</tr>
<tr>
<td>b</td>
<td>iswalnum</td>
</tr>
<tr>
<td>c</td>
<td>strcoll</td>
</tr>
<tr>
<td>d</td>
<td>nl_langinfo</td>
</tr>
<tr>
<td>e</td>
<td>fputwc</td>
</tr>
<tr>
<td>f</td>
<td>mbstowcs</td>
</tr>
<tr>
<td>g</td>
<td>wescat</td>
</tr>
<tr>
<td>h</td>
<td>iconvopen</td>
</tr>
<tr>
<td>i</td>
<td>catopen</td>
</tr>
<tr>
<td>j</td>
<td>regcomp</td>
</tr>
<tr>
<td>k</td>
<td>fimatch</td>
</tr>
</tbody>
</table>

Table 1. Internationalized Library Functions

2.2. Objectives

Previous work on internationalization like OSF/118N only considered a codeset with state-independent encoding scheme. Thus, it bypasses most internal problems in multibyte support. BSD/118N project is developed not only for its commercial purpose, but also for evaluating