Sorting Utility for Nepali in Linux

Bal Krishna Bal, Laxmi Prasad Khatiwada, Paras Pradhan, Pawan Chitrakar, Srishtee Gurung
Madan Puraskar Pustakalaya
bal@mpp.org.np,laxmi@mpp.org.np,paras@mpp.org.np, pawan@mpp.org.np,
srishtee@mpp.org.np

Abstract

This document is an analytical report on the sorting utility for Nepali in Linux, based on the research work conducted with the objectives of explaining how the available sorting utility in Linux works with Nepali characters and strings. Apart from the discussion of the core concepts required for sorting like collation and the different levels of comparison of strings employed for this purpose, the report includes several Nepali words and characters grouped according to different cases and thereafter looked how they are sorted. Exceptional cases are also presented.

1. Introduction

Sorting has always been an important aspect of computation. And unsurprisingly, this applies equally well to the Nepali Language Computing. Fortunately with the locale set to the Nepali locale file ne_NP, the sorting utility for Nepali in Linux is well supported and self-sufficient. This report would hence basically be focused on justifying this facility in Linux with the help of illustrative examples. In this regard, our hypotheses would be that the sorting utility in Linux works equally well with Nepali characters and strings and that any changes or modifications are not necessary to make in it so as to specifically deal with Nepali language except for the setting of the Nepali locale.

2. Methods

The methodology employed for the purpose of the study is purely a research-oriented one. So as to get convinced with the applicability of the existing sorting utility in Linux for Nepali as well, the sorting mechanism employed for collation in the Nepali locale file ne_NP was analyzed. This included the study of the sorting rule(s) used for collation followed by tests for the consistency of the sorting mechanism that considered several words grouped under separate cases.

3. Discussion

The Nepali locale file ne_NP contains a category LC_COLLATE that lists the Unicode characters from the Devanagri script in a sequential order. The formation of the three conjuncts - <ksha>, <tra> and <jna> from other Unicode characters are predefined using the collating-element statement, which precedes the list. These conjuncts appear later in the list as other normal characters. The extract of the LC_COLLATE category from the Nepali locale file ne_NP is presented below [1].

```
LC_COLLATE
collating-element <ksha> from
"<U0915><U094D><U0937>"
collating-element <tra> from
"<U0924><U094D><U0930>"
collating-element <jna> from
"<U091C><U094D><U091E>"

order_start forward [2]

<U0901>
<U0902>
<U0903>
<U0905>
<U0906>
<U0907>
<U0908>
<U0909>
<U090A>
<U090B>
<U090F>
<U090E>
<U0913>
<U0914>
<U0915>
<U0916>
<U0917>
```

```
<U0918>
<U0919>
<U091A>
<U091B>
<U091C>
<U091D>
<U091E>
<U091F>
<U0920>
<U0921>
<U0922>
<U0923>
<U0924>
<U0925>
<U0926>
<U0927>
<U0928>
<U092A>
<U092B>
<U092C>
<U092D>
<U092E>
<U092F>
<U0930>
<U0932>
<U0935>
<U0936>
<U0937>
<U0938>
<U0939>
<ksha>
<tra>
<jna>
<U093C>
<U093D>
<U093E>
<U093F>
<U0940>
<U0941>
<U0942>
<U0943>
<U0947>
<U0948>
<U094B>
<U094C>
<U094D>
<U0950>
<U0951>
<U0952>
<U0953>
<U0954>
<U0964>
<U0965>
<U0966>
<U0967>
<U0968>
<U0969>
<U096A>
<U096B>
<U096C>
<U096D>
<U096E>
<U096F>
<U0970>
order_end
END LC_COLLATE
```

The categorization defines character or string collation information. By collation, we understand the process and function of determining the sorting order of strings of characters [3].

The collation element, which is the unit of comparison for collation may be a character or a sequence of characters. Every collation element in the locale is entitled a set of weights. These weights determine if the collation element collate before, equal to, or after the other collation elements in the locale. The collation weights are used by applications programs that compare strings. Comparision of strings takes place by comparing the collation weights of each character in the string until we end up with a difference or that the strings are found to be equal. The comparision may occur multiple times if there are multiple collation orders defined in the locale. For instance, French makes use of secondary collation weights if the primary collation weights are found to be equal [1].

The sorting rule that the Nepali locale file ne_NP employs is the "forward rule". This is evident from the keyword "forward" that follows the keyword "order_start" in the LC_COLLATE category. This means that sorting occurs taking into account the order of characters sequence as it appears in the list of characters inside the LC_COLLATE category in between the keywords "order_start" and "order_end".

The Unicode Collation Algorithm takes an input Unicode string and a Collation Element Table. This table contains the mapping data for characters. A sort key is produced, which is an array of unsigned 16-bit integers. These sort keys may then be binary-compared to give the correct comparison between the strings. By default, the algorithm uses three fully-customizable levels. For the Latin script, these levels are [4]:

1. alphabetic ordering
2. diacritic ordering
3. case ordering.

If the distinction cannot be evaluated while comparing strings, a final level for tie-breaking may be used.

Hence, each character has a collating element that has at most four different weights, namely primary, secondary, tertiary and quaternary. The latter weights are ignored if differences are noted in the primary weights itself. This pattern holds true for other weights successively.

In the case of Nepali words basically, "the basic character or the alphabets" are looked while sorting. In other words, alphabetic ordering is employed. For example, with the words □□□, खरायो, and गम□□□र the weights other than primary are ignored and collating occurs on the basis of the primary weights since each of these words differ from each other with the first letter itself. Besides, diacritic ordering is also employed that deals with words containing the next three characters [5]:

- Devanagari sign CANDRABINDU;
- Devanagari sign ANUSVARA;
- Devanagari sign VISARGA.

In the ordered list of characters as defined inside the LC_COLLATE category in the Nepali locale file ne_NP, these characters take the first three positions. A diacritic mark added to a letter indicates a special pronunciation. Apart from these two levels of ordering, exceptions also prevail in terms of sorting in Nepali. Such exceptions are basically due to the different writing systems that are employed to write the same word in Nepali and Sanskrit.

4. Results

A test of the normal functionality or the applicability of the sorting utility in Linux for Nepali was accomplished with the aid of several group of Nepali words and characters taken into consideration under separate cases. Results of the test are as presented below:

Before sorting	After sorting
ज्ञानी	क्षेत्रीय
क्षेत्रीय	त्रिकोण
त्रिकोण	ज्ञानी

Table 1: Case 1 (Conjuncts)

Before sorting	After sorting
कमल	कमल
जरायो	खरायो
खरायो	खल्ति
नमस्ते	जरायो
दराज	दराज
खल्ति	नमस्ते

Table 2: Case 2(Consonants)

Before sorting	After sorting
विचार	कीरा
कीरा	फु ल
फुल	फू ल
फूल	बगर
वंश	बाकस
बगर	वंश
बाकस	विचार

Table 3: Case 3 (Dependent vowels)

Before sorting	After sorting
क्	क
क	क्
क्म	क्म

Table 4: Case 4 (Equivalent Forms)

Before sorting	After sorting
जांगर	जाँगर
जाःगर	जांगर
जाँगर	जाःगर

Table 5: Case 5 (Diacritic marks)

4.1. Test cases and exceptions

While the sorting mechanism for the first three cases is apparent, the fourth case requires some further explanation. Although, the two characters क and क् are almost equivalent phonetically, the sorting utility sorts them as the normal क preceding the क्. This accounts to the character ◌् existing in क्, but absent in क. The code of the character ◌् appears in the order list a lot later which implies that it has a smaller weight. This indeed justifies the sorting of क prior to क्. Some of the words looked upon under case 5.

"Diacritic marks" do not grammatically exist but are specifically included for checking the diacritic ordering. Exception to the alphabetical and diacritic ordering in terms of sorting hold some typical Nepali words like ब्रह्मा, which is written as ब+ ◌्+र+ ह+ ◌् +म+ ◌ा though the character ह precedes म, sorting is guided by the character म rather than the character ह. A similar situation is in the case of ब्राह्मण, which though written as ब + ◌् +र+ ◌् + ◌ा +ह+ ◌् +म+ण with the character ह before म, sorting is guided by the character म rather than the character ह . It is found that the words ब्रह्मा and ब्राह्मण are writing forms in Sanskrit, while the same words in Nepali version are written as ब्रम्हा and ब्राम्हण. In practice, however according to linguists, both of the writing forms of these words equally co-exist.

Despite of all the good aspects of the sorting utility for Nepali in Linux, a warning message of the following character is displayed when after the locale is set to ne_NP and some applications like Mozilla-FireFox is launched. The warning message is:

sed -e expression #1: char 12: Mozilla -FireFox Invalid Collation character. The possible remedies to the problem are being searched at the moment. While the warning does not have major impacts on the working of the sorting utility, still its causes are worth exploring.

5. Conclusion

The results obtained from the research work show that the sorting utility available in Linux, leaving aside a few nominal exceptions works well with Nepali words and characters. This implies that apart from setting the locale to the Nepali locale file ne_NP, there is not the necessity for making any modification in the source code itself.

6. References

For preparing this document the following links and documents were referred to.

[1] http://news.software.ibm.com

[2] http://www.mpp.org.np/detail_guide/fonts/Samanata.ttf

[3] http://www.unicode.org

[4] http://www.benutzer.de/?content=610&

[5] http://ra.dkuug.dk/jtc1/sc2/wg2/docs/n1675.pdf

[6] "Nepalinux "http://www.nepalinux.org/ldf/ne_NP

[7] "Linux and Localization Tutorial" http://www.nepalinux.org/linuxtutorial.ppt

[8] http://www.publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/files/aixfiles/LC_COLLATE.htm

[9] http://www.unicode.org/reports/tr32/tr32-6.html