Document number: SC22/WG14 N659
(J11 97-022)

Title: toups/tolows

Author: Keld Simonsen
Author affiliation: DKUUG
Postal address: Fruebjergvej 3, DK-2100 København Ø
Email address: keld@dkuug.dk
Telephone number: +45 3917-9944
Fax number: +45 3325-6543

Sponsor: DS
Date: 1997-02-09

Proposal category:
    __ Editorial change/non-normative contribution
    __ Correction
    XX New feature
    __ Addition to obsolescent feature list
    __ Other (please specify)

Area of standard affected:
    XX Environment
    __ Language
    __ Preprocessor
    XX Library
       XX Macro/typedef/tag name
       XX Function
       XX Header
    __ Other (please specify)

Prior art: partly implemented in some compilers
Target audience: general

Related documents: none
Proposal attached: proposal paper
Review committeee: Keld Simonsen, Frank Farance

Abstract:

The paper gives proposals to solve a cultural problem with German double-s, alongside with introducing string functions in use a number of places.

The proposer is neither German nor Dutch, and apologizes for any errors in the cultural descriptions.

Introduction

This paper gives proposals for solving the problem with upper-casing the letter frequently used in german text, the LATIN LETTER SHARP S. Also problems with uppercasing the Dutch IJ letter will be addressed. The problem with the current semantics of the C standard is that the toupper() and tolower() functions only can output one character, where it is culturally needed to output more than one. Also there is a problem that it is culturally needed that two characters in a row be uppercased together, and the functionality proposed can also address this.

The two cases that are addressed are:

1. German LATIN LETTER SHARP S are converted from lowercase to uppercase to two LATIN CAPITAL LETTER S. Example: Maß (measure). New German orthography rules have been introduced (Duden 1996), where the ß has been removed. Still my impression is that the old rules are in widespread use.

2. In Dutch the sequence "ij" is always uppercased together to the sequence "IJ". This is because the Dutch consider the "ij" ("laange ij") as some kind of single unit (maybe a letter). In examples where there should be a capitalization of the initial letter, as at the beginning of a sentence, there is a need to do this together. Example: "IJs is lekker" (Icecream is wonderful). Sometimes the "IJ" is written as one character Ĳ and the it is doable with current specifications, but most of the times it is written with two characters.

The paper describes a proposal and some alternatives.

Proposal

The proposal is to intotroduce two functions, that operate on a string of characters, and return a string of characters. These functions then should be available in char, widechar and multibyte forms. The functions to convert a string to uppercase or lowercase is very commonly found in compilers and application code.

The functions to add are for the char type handling:

int toups(char *s1, char *s2, int n);
Converts string "s1" to string "s2" using the uppercasing information of the current locale. One character in the input string may be converted to more than one character in the output string. At most "n" characters are converted. The function returns "−1" if the input string cannot be converted within "n" characters, otherwise it returns the number of characters in the output string.

int tolows(char *s1, char *s1, int n);
Similar to toups() but actually not needed for the German problem, and only included for convenience and consistency. Many applications convert a string to lowercase representation before doing further processing.

To implement this proposal in the C standard text, it is needed to add the description in the library clause, and also add definition headers to "string.h" and othe include files. There may be a need to also write extra information for the locale structures so that the touppr/tolower definitions can have more than one character as result. Alternatively a new set of definitons may be introduced.

The Dutch problem has not been handled here as it was not a part of the original problem, and as the problem only occurs when the initial "letter" is to be uppercased, for which there is no traditional functionality being employed in applications. If all letters are uppercased, there is no problem with the current specifications.

Alternate naming is strtoupper() and strtolower().

Alternatives

Widechar intrinsic characters

When the processing is done as widechar characters, intrinsic characters may be defined to represent the character sequences

two LATIN CAPITAL LETTER S characters
the character LATIN CAPITAL LETTER I and LATIN CAPITAL LETTER J
the character LATIN SMALL LETTER I and LATIN SMALL LETTER J

It is possible with current POSIX charmap specifications conforming to ISO/IEC 9945-2:1993 to then define mappings between the intrinsic widechar representation and the external coded character representation.

The problem with this way of solving the issue is that it may be difficult to get this specified in locale and charmap specifications, and that it may be culturally incorrect. For example rhe sequence "SS" may not always be convertible to "ß" in German, and in other languages, and the sequence "ij" may not be considered a single unit in many languages, although the consequenses of considering it a single unit may not be distinguashable. These problems may be reduced by having separate POSIX charmaps for use with German or Dutch environments.

The intrinsic character representation solution seems to not be causing semantic problems for the Dutch issue, and may thus be recommended.

Initial uppercasing function

To solve the Dutch problem, a function converting the initial "unit" of a string could be introduced in the standard. The interface could be:

```
int initup(char *s1, char *s2, int n);
```

<END OF DOCUMENT>