Sắp xếp (sort) string Tiếng Việt

written by HVN on 2017-06-06

Một vấn đề không hề mới, xưa như năm 1991, và có vẻ đâu đó đã được giải quyết 😒

Vậy nhưng nếu thử hỏi một lập trình viên sử dụng ngôn ngữ lập trình bất kỳ hãy viết một đoạn code sắp xếp những cái tên sau theo thứ tự của tiếng Việt:

```
In [14]: provinces = ['Hải Dương',
    ...:  'Hưng Yên',
    ...:  'Hà Nội',
    ...:  'Hải Phòng',
    ...:  'Hậu Giang',
    ...:  'Hòa Bình',
    ...:  'Hà Nam',
    ...:  'Hà Giang',
    ...:  'Hà Tĩnh']
```

Kết quả thu được sẽ khá bất ngờ:

```
In [15]: sorted(provinces)
Out[15]:
['Hà Giang',
 'Hà Nam',
 'Hà Nội',
 'Hà Tĩnh',
 'Hòa Bình',
 'Hưng Yên',
 'Hải Dương',
 'Hải Phòng',
 'Hậu Giang']
```

Lẽ ra, thứ mà chúng ta muốn nhìn thấy phải là "Hưng Yên" đứng cuối list, khi mà chữ "ư" đứng sau tất cả các phụ âm khác.

Vậy function `sorted` có bug?

Hãy thử xem codepoint của từng chữ cái thứ 2 trong mỗi từ:

```
In [19]: [(s[1], ord(s[1])) for s in sorted(provinces)]
Out[19]:
[('à', 224),
 ('à', 224),
 ('à', 224),
 ('à', 224),
 ('ò', 242),
 ('ư', 432),
 ('ả', 7843),
 ('ả', 7843),
 ('ậ', 7853)]
```

Vậy `sorted` hoàn toàn không sai, nó sắp xếp các chữ cái theo thứ tự các xuất hiện trong bảng mã Unicode (codepoint). Chữ "ư" nằm ở vị trí 432, đứng trước chữ "ậ" với vị trí 7853.

`sorted` đã đúng, nhưng đây không phải cách sắp xếp chúng ta mong muốn.

Vậy điều gì ảnh hưởng đến thứ tự khi sắp xếp của các chữ cái? Rõ ràng nếu theo chuẩn của người Mỹ, người ta sẽ sắp xếp như trên, nhưng trong bảng chữ cái tiếng Việt, thứ tự lại khác. Những thứ liên quan đến ngôn ngữ địa phương như vậy được biểu diễn lên máy tính bằng khái niệm "locale".

Locale là gì /ləʊˈkɑːl/

- English: a place where something happens or is set, or that has particular events associated with it
- Vietnamese: mội nơi mà điều gì đó xảy ra hoặc có những sự kiện gắn liền với nó.
- Một chương trình hỗ trợ locale là chương trình tôn trọng sự lựa chọn dựa trên nền tảng văn hoá của người dùng, như bảng chữ cái, cách sắp xếp và cách viết số (người Việt Nam thường dùng dấu `,` trong số thực: 1,5. Người châu Mỹ lại dùng dấu `.`: 1.5) (Dịch theo tài liệu của Postgresql)
- Một locale là một bộ luật ngôn ngữ và văn hoá. (theo manpage của Ubuntu)

Locale không chỉ là khái niệm của riêng ngôn ngữ lập trình nào mà đòi hỏi là hỗ trợ của toàn hệ thống. Các hệ điều hành (OS) đều hỗ trợ đặt các và sử dụng locale khác nhau.

Cách biểu diễn một locale

Một locale thường được biểu diễn ở dạng "ngônngữ_quốcgia":

- `en_US` (U.S. English)
- `fr_CA` (French Canadian)
- `vi_VN` (Vietnamese Vietnam)

Trên hầu hết các hệt thống UNIX (Ubuntu, OSX ...), có thể liệt kê các locale đang hỗ trợ trên máy bằng lệnh `locale -a`:

```
$ locale -a | head -n3
C
C.UTF-8
en_AG
$ locale -a | tail -n3
POSIX
vi_VN
vi_VN.utf8
```

Các locale có thể được cài thêm, ví dụ trên Ubuntu 16.04, locale Việt Nam không được cài sẵn, phải cài thêm package `language-pack-vi` để có locale vi_VN:

```
apt-get install -y language-pack-vi
```

Nếu có nhiều hơn một bộ ký tự cho một locale, có thể chỉ rõ ra nó ở dạng: "ngônngữ_quốcgia.bộkýtự": Ví dụ:

- vi_VN.tcvn (Tiêu chuẩn Việt Nam, hay VNS)
- vi_VN.utf8
- vi_VN.vscii (Cho ngang ngửa với ASCII của Mỹ 😁)

Danh sách một số locale phổ biến hỗ trợ bởi GCC, hay danh sách đầy đủ của OS, hay trong code của module `locale` của CPython

Khi không có locale nào được set, locale mặc định là C hay POSIX.

Locale categories

Đôi khi, người ta lại muốn dùng bảng chữ cái của Việt Nam, nhưng đơn vị tiền tệ của Mỹ, thời gian theo format của Anh, những nhu cầu lẫn lộn này được đáp ứng bởi locale sẽ chia nhỏ thành "locale subcategories" và mỗi category sẽ điều chỉnh một khía cạnh của luật locale:

- LC_COLLATE: thứ tự sắp xếp string /kəˈleɪt/, ảnh hưởng đến các function strcoll, strxfrm
- LC_CTYPE: phân nhóm chữ cái (đâu là một chữ cái, chữ cái dùng 1byte hay nhiều bytes? chữ viết hoa tương ứng của mỗi chữ là chữ nào?)
- LC_MESSAGES: ngôn ngữ của message
- LC_MONETARY: format (định dạng) của tiền
- LC_NUMERIC: format của số
- LC_TIME: format của ngày tháng
- LC_TELEPHONE: format số điện thoại
- ... còn nhiều
- LANG: default cho các LC_*
- LC_ALL: tất cả các category nói trên.

Thứ tự xử lý locale

Các locale category thường được set làm biến môi trường.

- Nếu `LANG` không null thì giá trị của LANG được sử dụng làm default cho tất cả các giá trị không được set / null khác.
- Nếu `LC_ALL` được set, không null, thì giá trị của LC_ALL sẽ được sử dụng, thay cho tất cả các giá trị khác.
- Nếu biến nào được set không null thì biến đó được sử dụng. Vậy mức ưu tiên là: `LC_ALL > LC_* > LANG`

Theo http://manpages.ubuntu.com/manpages/xenial/en/man1/locale.1posix.html

Set/get locale trên Python

```
In [1]: import locale


In [3]: locale.getlocale(category=locale.LC_CTYPE) # LC_TYPE là category mặc định
```

```
Out[3]: ('en_US', 'UTF-8')

In [4]: locale.getlocale()
Out[4]: ('en_US', 'UTF-8')

In [5]: locale.setlocale(locale.LC_ALL, 'vi_VN')
---------------------------------------------------------------------------
Error                                     Traceback (most recent call last)
<ipython-input-5-7ee8baa2ff04> in <module>()
----> 1 locale.setlocale(locale.LC_ALL, 'vi_VN')

/Users/hvn/python3/lib/python3.5/locale.py in setlocale(category, locale)
    593         # convert to string
    594         locale = normalize(_build_localename(locale))
--> 595     return _setlocale(category, locale)
    596
    597 def resetlocale(category=LC_ALL):

Error: unsupported locale setting
# Do máy không có locale vi_VN

# set locale cho tất cả category sử dụng giá trị mặc định (thường lấy trong b
iến môi trường LANG)
In [6]: locale.setlocale(locale.LC_ALL, '')
Out[6]: 'en_US.UTF-8'

# Trả về database của các convention hiện tại
In [7]: locale.localeconv()
Out[7]:
{'currency_symbol': '$',
 'decimal_point': '.',
 'frac_digits': 2,
 'grouping': [3, 3, 0],
 'int_curr_symbol': 'USD ',
```

```
 'int_frac_digits': 2,
 'mon_decimal_point': '.',
 'mon_grouping': [3, 3, 0],
 'mon_thousands_sep': ',',
 'n_cs_precedes': 1,
 'n_sep_by_space': 0,
 'n_sign_posn': 1,
 'negative_sign': '-',
 'p_cs_precedes': 1,
 'p_sep_by_space': 0,
 'p_sign_posn': 1,
 'positive_sign': '',
 'thousands_sep': ','}


# Đổi locale sang C, ở locale này không có đơn vị tiền tệ.
In [15]: locale.setlocale(locale.LC_ALL, 'C'); locale.localeconv()['currency_
symbol']
Out[15]: ''


# Khi đổi lại về locale default (thường lấy từ LANG), ở đây là en_US.UTF8, ký
hiệu đơn vị tiền tệ là $.
In [16]: locale.setlocale(locale.LC_ALL, ''); locale.localeconv()['currency_s
ymbol']
Out[16]: '$'
```

Trên một máy có locale vi_VN, có thể thay đổi và xem đơn vị tiền tệ:

```
In [2]: locale.setlocale(locale.LC_ALL, 'vi_VN')
Out[2]: 'vi_VN'


In [3]: locale.localeconv()
Out[3]:
{'currency_symbol': '₫',
 'decimal_point': ',',
 'frac_digits': 0,
```

```
 'grouping': [3, 3, 0],
 'int_curr_symbol': 'VND ',
 'int_frac_digits': 0,
 'mon_decimal_point': ',',
 'mon_grouping': [3, 3, 0],
 'mon_thousands_sep': '.',
 'n_cs_precedes': 1,
 'n_sep_by_space': 0,
 'n_sign_posn': 1,
 'negative_sign': '-',
 'p_cs_precedes': 0,
 'p_sep_by_space': 0,
 'p_sign_posn': 1,
 'positive_sign': '',
 'thousands_sep': '.'}
```

Chú ý việc set này chỉ có tác dụng tại phiên làm việc hiện thời. Để thay đổi locale của toàn hệ thống (Ubuntu), phải thay đổi trong `/etc/default/locale`

```
root@hvn:~# cat /etc/default/locale
LANG="en_US.UTF-8"
```

rồi chạy `locale-gen`.

Sắp xếp string theo kiểu Việt Nam

Nếu đã đọc đến đây và hiểu những gì viết trong bài thì bạn sẽ biết rằng thứ tự các chữ cái khi sắp xếp phụ thuộc vào locale category: `LC_COLLATE`

```
In [1]: import locale


In [2]: locale.getlocale()
Out[2]: ('en_US', 'UTF-8')


In [3]: provinces = ['Hải Dương',
   ...:      ...:  'Hưng Yên',
```

```
   ...:     ...:  'Hà Nội',
   ...:     ...:  'Hải Phòng',
   ...:     ...:  'Hậu Giang',
   ...:     ...:  'Hòa Bình',
   ...:     ...:  'Hà Nam',
   ...:     ...:  'Hà Giang',
   ...:     ...:  'Hà Tĩnh']

In [4]: sorted(provinces)
Out[4]:
['Hà Giang',
 'Hà Nam',
 'Hà Nội',
 'Hà Tĩnh',
 'Hòa Bình',
 'Hưng Yên',
 'Hải Dương',
 'Hải Phòng',
 'Hậu Giang']

In [5]: locale.setlocale(locale.LC_COLLATE, 'vi_VN')
Out[5]: 'vi_VN'

In [6]: sorted(provinces, key=locale.strxfrm)
Out[6]:
['Hà Giang',
 'Hải Dương',
 'Hải Phòng',
 'Hà Nam',
 'Hà Nội',
 'Hà Tĩnh',
 'Hậu Giang',
 'Hòa Bình',
 'Hưng Yên']
```

```
In [11]: locale.strxfrm?
Docstring:
strxfrm(string) -> string.

Return a string that can be used as a key for locale-aware comparisons.
Type:      builtin_function_or_method
```

Giờ thì Hưng Yên đã đứng cuối list 😍.

- PS: Chọn locale lúc tạo database là điều rất quan trọng cần chú ý (PostgreSQL, MySQL).
- PS2: khi chạy chương trình trên server, nhớ set locale: http://www.familug.org/2017/03/phai-set-locale-trong-upstart.html
- PS3: với Python có thể tham khảo sử dụng thư viện `pyicu`

Tham khảo

- Postgresql doc: https://www.postgresql.org/docs/9.3/static/locale.html
- Sorting howto: https://docs.python.org/3/howto/sorting.html#odd-and-ends
- Locale stdlib: https://docs.python.org/3/library/locale.html

Hết. HVN at http://www.familug.org/ and http://pymi.vn