

I18n กับการพัฒนา software ของประเทศไทย

นำเสนอโดย

นายชัยวัฒน์ มณีบุญ

รหัส 45033967 ภาควิชา วิทยาการคอมพิวเตอร์

มหาวิทยาลัยบูรพา

อาจารย์ที่ปรึกษา อ.ธวัชชัย เอี่ยมไพโรจน์

Overview

- ความหมายของ i18n
- ความเข้าใจเกี่ยวกับ Software Localization และ Thai Localization
- Character set encoding
- Locale Technology and Categories
- Localization
- ตัวอย่างโปรแกรมกับ i18n

i18n คืออะไร?

- **Internationalization (aka. i18n)** คือการที่ซอฟต์แวร์สนับสนุนการนำไปใช้ในประเทศ ภาษา และวัฒนธรรมต่างๆ ที่แตกต่างกันได้ เช่น การเข้ารหัสอักขระ รูปแบบวันที่ ตัวเลข อินเทอร์เน็ตเฟสภาษาต่างๆ เป็นต้น โดยเขียนโปรแกรมเพียงครั้งเดียว และใช้ไฟล์ไบนารีที่คอมไพล์แล้วตัวเดียวกันได้ทันที โดยไม่ต้องแก้ไขแล้วนำไปคอมไพล์ใหม่

มาตรฐานของ i18n

- แสดง **character** ของภาษาแต่ละท้องถิ่นได้
- นำเข้า **character** ของภาษาแต่ละท้องถิ่นได้
- จัดการกับ **character** ของภาษาแต่ละท้องถิ่นได้
- แสดง **Messages** ของโปรแกรมในระบบภาษาท้องถิ่นได้
- จัดการ **input** และ **output** ของตัวเลข,วันที่,เงิน ของท้องถิ่นได้
- **Sort** และ จัดประเภท ของ **character** ในภาษาแต่ละท้องถิ่นได้

L10N,I18N,M17N model

■ L10N model

- สามารถที่จะจัดการกับ 2 ภาษา หรือ character code (ASCII) กับอีก 1 character code
- ตัวอย่าง software (Emacs)

■ I18N model

- สามารถที่จะจัดการได้หลายๆภาษา กับอีก 1 ภาษาในเวลาเดียวกัน โดยการ set LANG environment variable ของระบบ

■ M17N model

- สามารถที่จะจัดการกับหลายๆ ภาษาในเวลาเดียวกัน
- ตัวอย่าง software (XEmacs)

ความเข้าใจเกี่ยวกับ **Software Localization** และ **Thai Localization**

■ **Software Localization:**

หมายถึง การแปลอินเทอร์เฟซของซอฟต์แวร์ จากภาษาหนึ่ง ไปเป็นภาษาท้องถิ่น เพื่อให้ผู้ใช้ภาษาท้องถิ่นนั้นๆ สามารถเข้าใจและใช้ ซอฟต์แวร์ได้ง่ายขึ้น

■ **Thai Localization:**

หมายถึง การแปลอินเทอร์เฟซของซอฟต์แวร์ จากภาษาหนึ่ง (ในปัจจุบันหมายถึงภาษาอังกฤษ) มาเป็นภาษาไทย

Locale Technology

- หมายถึง การใช้ความสามารถของบาง ฟังก์ชันในภาษา C ซึ่งจะขึ้นอยู่กับการ **set LOCALE environment** ของระบบ โดยสามารถที่จะกำหนดให้โปรแกรมใช้คุณสมบัตินี้ได้โดยใช้ **function setlocale()**
 - **Setlocale(LC_TIME, "de_DE");**

Locale Categories

- **LC_CTYPE** รหัสอักขระที่ใช้
- **LC_NUMERIC** รูปแบบการแสดงผลตัวเลข เช่น 1,200.50 หรือ 1.200,50
- **LC_TIME** รูปแบบการแสดงผลวันที่/เวลา เช่น 12 เม.ย. 2548
- **LC_COLLATE** การเปรียบเทียบสตริงของตัวอักษร ทำให้การเรียงลำดับถูกต้อง
- **LC_MONETARY** รูปแบบการแสดงผลตัวเลขทางการเงิน เช่น 100.00฿
- **LC_MESSAGES** ภาษาของข้อความในอินเทอร์เน็ตเฟส เช่น เมนู หรือ กรอบข้อความ

Locale Categories(ต่อ)

- **LC_PAPER** ขนาดกระดาษมาตรฐาน เช่น "A4"
- **LC_NAME** รูปแบบการแสดงชื่อ เช่น "บุญมี ทองมาก", "Asimov, Isaac"
- **LC_ADDRESS** รูปแบบการแสดงที่อยู่
- **LC_TELEPHONE** รูปแบบการแสดงผลหมายเลขโทรศัพท์ เช่น 0-2564-6900
- **LC_MEASUREMENT** หน่วยวัดมาตรฐาน เช่น แบบอังกฤษ (นิ้ว) หรือ ISO (เซนติเมตร)
- **LC_IDENTIFICATION** รูปแบบหมายเลขบัตรประจำตัว

Limit of Locale technology

- Locale model ไม่สามารถที่จะจัดการกับ 2 locale ในเวลาเดียวกัน

Thai character set encoding

- 8-bits encoding: TIS-620, ISO-8859-11, IBM-CP874, Windows-874, mac-thai
- Unicode 16-32 bits: UTF-8, UTF-16, UTF-32

รูปแบบของ locale

- th_TH.TIS-620
 - th = ภาษาไทย
 - TH = ประเทศไทย
 - TIS-620 = รหัสอักขระที่ใช้

การกำหนด locale

- กำหนดทั้งระบบสำหรับ Red Hat/Fedora/LinuxTLE
 - /etc/sysconfig/i18n
 - ❖ LANG=th_TH.TIS-620
- กำหนดชั่วคราวสำหรับการรันแอปพลิเคชันแต่ละครั้ง
 - \$ export LC_MESSAGES=en_US
 - \$ gcalctool
 - \$ LC_MESSAGES=en_US gcalctool

การเขียนโปรแกรมให้ **Localize** ได้

- ใช้ฟังก์ชัน `setlocale()` เพื่อกำหนด `locale`
- ใช้ `library` ชื่อ `gnu gettext` ซึ่งเป็นมาตรฐานที่ใช้กันทั่วไปในซอฟต์แวร์ โอเพนซอร์ส
- ใช้ฟังก์ชัน `textdomain()` เพื่อกำหนดชื่อ `textdomain` ที่ต้องการใช้ ซึ่งชื่อที่กำหนด จะถูกใช้เป็นชื่อไฟล์ `.mo` ด้วย
- ใช้ฟังก์ชัน `gettext()` กับ `string` ทั้งหมดที่ต้องการแสดงผลแบบให้สามารถนำไปแปลได้

ตัวอย่าง **source code** โปรแกรม

- Hello.c

การสร้าง และจัดการไฟล์ po

- การสร้าง template ของ po (ไฟล์ .pot)
 - `xgettext --keyword=_ --flag=_:1:pass-c-format -o po/hello-c.pot hello.c`
 - `cd po`
 - `cp hello-c.pot th.po`
- การแปลข้อความ สามารถใช้โปรแกรม `poedit`, `kbabel`, `gtranslator` เปิดไฟล์ po ขึ้นมาแปลได้
 - `$ gtranslator th.po`

การสร้าง และจัดการไฟล์ **po**(ต่อ)

- การแปลง **po** เป็น **mo**
 - `$ msgfmt -o th.mo th.po`
 - `$ cp th.mo /usr/share/locale/th/LC_MESSAGES/hello-c.mo`
- การ **merge** ข้อความที่แปลแล้ว กับไฟล์ **pot** ใหม่ เมื่อข้อความในซอร์สมีการเปลี่ยนแปลง หรือเพิ่ม
 - `$ mv th.po th.po.old`
 - `$ msgmerge th.po.old hello-c.pot > th.po`

สรุป

- การเขียนโปรแกรมให้เป็น **standard** ต้องไม่ยึดติดกับภาษาใดภาษาหนึ่ง
- **I18n** เป็นมาตรฐานเพื่อให้สามารถเขียนโปรแกรมให้เป็น **Internationalization**

Reference

- <http://www.debian.org/doc/ddp>
- <http://www.faqs.org/rfcs/>
- <http://www.unicode.org>
- <http://www.nectec.or.th>

คำถาม