GNU/Linux Localization to Persian Language

Mohammad Khansari,
Director of
Iran GNU/Linux Localization Project

Advanced Information and Communication Technology Center
of Sharif University, Iran
Khansari@aictc.com

First International Conference on IT-based localization in the area of script and language, ITRC, Tehran, Iran
May 15-16, 2005

Outline

- GNU/Linux localization process
- History of Persian GNU/Linux localization Project
- Goals of the project
- Sub-projects
- Cultural readiness for GNU/Linux
- References

Linux localization

There are many different methods used to localize GNU/Linux, using different encoding, input and display systems. Present, the most technically effective method is localization via the ***Linux-Unicode-OpenType*** model.

Unicode?

- The Unicode is an industry standard for encoding characters and symbols.
- It is closely related to the ISO Universal Character Set standard 10646.
- The Unicode Consortium, co-founded by Apple and Xerox in 1991, now has more than 100 members, including Adobe, IBM, Microsoft, Sybase, Compaq, Hewlett Packard, Oracle, Sun Microsystems, Netscape and Ericsson.
- The aim of Unicode and ISO 10646 is to encompass all of the languages of the world, with each character code corresponding to a ‘glyph’. Combinations of character codes produce combined glyphs for complex characters (particularly in the Asian languages).
- The initial Unicode standard specified an encoding for 16-bit characters, which allows for a total of 65,535 possible characters/symbols.
- Later versions of the standard have expanded the encoding to a 32-bit range, allowing over one million different characters and symbols to be encoded.
- It is the most relevant encoding system for the Internet.

OpenType?

- Fonts are at the ‘front end’ of localization and often receive the most attention from non-technical observers.
- OpenType font file formats is the appropriate standard for font development in localization efforts.
- OpenType is a cross-platform font file format jointly developed by Microsoft and Adobe.
- Opentype is based on the Unicode encoding standard and offers multiple language character sets in one font file. Whereas traditional Western Postscript fonts are limited to 256 glyphs, an OpenType font may contain more than 65,000 glyphs, allowing multiple languages to be displayed using a single font.

Localization steps

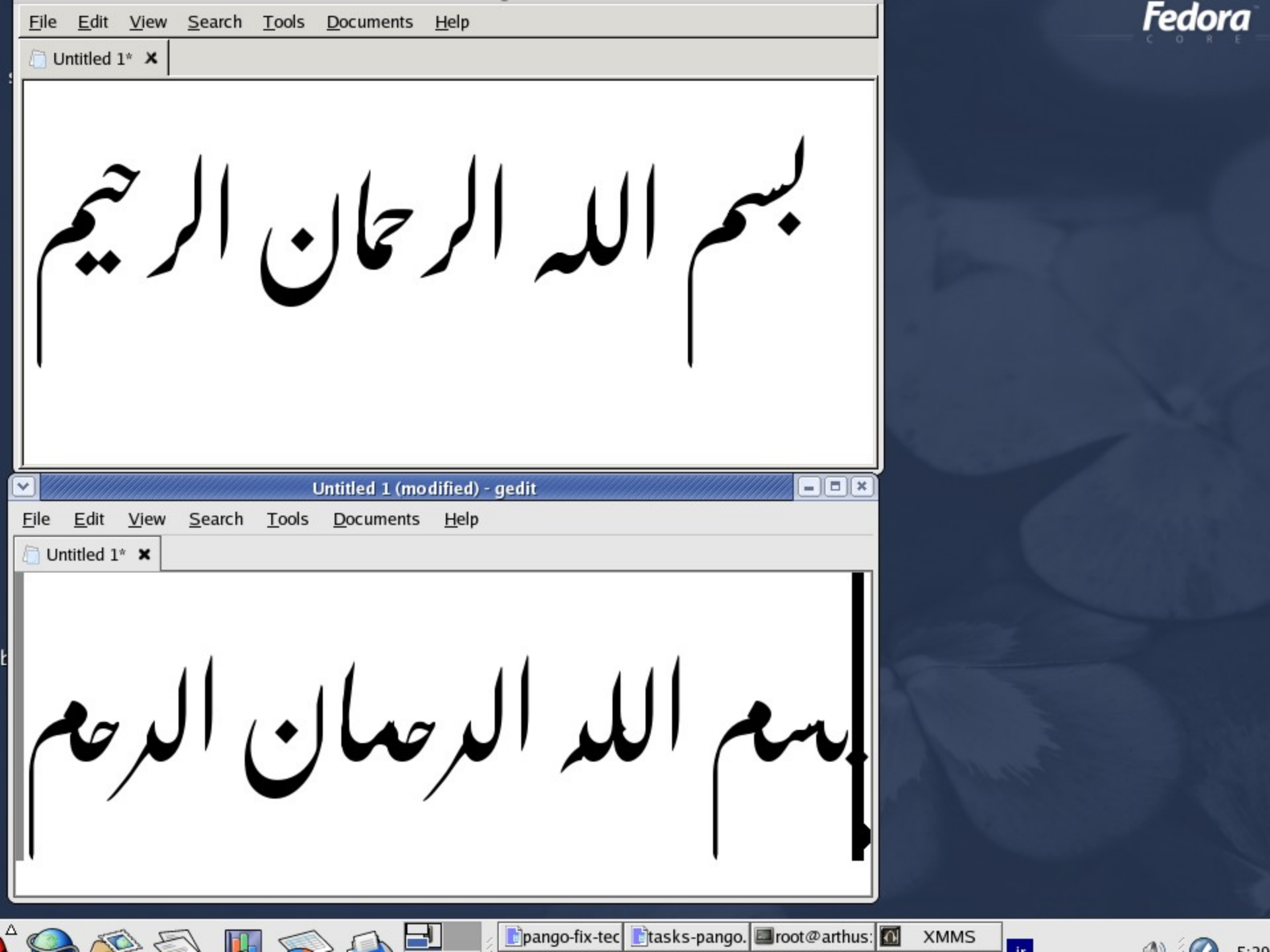
- Localization steps using the Linux-Unicode-OpenType model are as follows:
  1. Unicode standard corrections/enhancements
  2. Font development
  3. Input methods
  4. Locale development
  5. International contribution to modify main FOSS Libraries to handle outputs of the above steps
  6. International contribution to modify FOSS applications to handle outputs of the above steps
  7. Translating application messages
- Ensuring that all specification (steps 1-4) are accepted by the standard bodies both international and within the country
- Ensuring that all changes in steps 5-7 are accepted by the global FOSS community

Unicode standard corrections/enhancements

- Creating encoding that adequately handles the needs of the countless languages throughout the world is highly complex.
- The immensity of this task has resulted in errors and inadequacies in the specification of certain languages, particularly languages from countries that have low levels of ICT development.
- Additionally, while Unicode may have included encoding for all of the major languages in the world, encoding for the other languages and dialects (India alone has over 1,000 languages and dialects) is either incomplete or non-existent.
- In countries where the existing Unicode standard is lacking, a review of the existing Unicode standard and recommendation of changes to the Unicode Consortium will be necessary.
- Iran has passed this step and has an Unicode standard (ISIR 6219).

Font development

- After Unicode standard the next challenge is a freely available, cross-platform fonts.
- Without fonts, it is impossible to display, use and manipulate any language electronically.
- Modern fonts, particularly OpenType fonts, are more than just the visual representation of a language. OpenType fonts contain the logic behind the display of the words, how glyphs interact with and change surrounding glyphs.
- Languages that differ greatly from the western alphabet (e.g. Persian, Arabic, Pashto) often do not have a commonly available, non-proprietary font.
- Font development is not a small task. A high-quality, professional font can take several years to develop.
- We have contributed to improve international OpenType font rendering engine for Linux (pango) and developed a few fonts. Developing some other fonts is under progress.


File Edit View Search Tools Documents Help
Untitled 1*
بسم الله الرحمان الرحيم
Untitled 1 (modified) - gedit
File Edit View Search Tools Documents Help
Untitled 1*
بسم الله الرحمان الرحم
pango-fix-tec
tasks-pango.
root@arthus:
XMMS

Input methods

- The most common input method in computing is via the keyboard.
- Many countries have created mappings between the standard characters in their local language (often ad hoc). For example, there are several keyboard layouts in use regularly in Bangladesh.
- The lack of a single standard is a result of and contributes further to incompatible implementations of character sets/encoding, keyboard mappings, fonts, and the like.
- Once an input method has been standardized, software has to be written to implement the standard under GNU/Linux.
- We have done this step in Iran, We have a National Standard Keyboard map (ISIR 2901) which is used in Linux distributions. Nevertheless, the Windows Persian keyboard is not compatible with this standard.

Locale development

- While most major FOSS applications have been internationalized, some modification may still be necessary to adapt to local language characteristics. For example, most word processors break words on a space but in languages that do not use spaces, special rules must be created to specify breaking order. Similar problems exist with word sorting, text flow and other issues. Most languages will require minimal modification but certain languages may require extensive modification to applications.
- Additionally, locale-specific information such as date format, currency symbols and other issues has to be specified.
- We have defined a standard set of Locales for Iran.
- We have also contributed to design the latest Unicode compatible FreeBiDi algorithm and implementation (Freedesktop.org). The results are applicable to Arabic script too.

Translating application messages

- The next step in localizing GNU/Linux involves the translation of messages that the application passes to the user. Messages such as “File Not Found” or “Operation Complete” have to be translated to the local language.
- This task involves very little technical skill as the messages are normally stored in text files for easy viewing and editing. However, translating the thousands of messages and help files is an undertaking that can take several years to complete and is often the slowest part of the localization process.
- We are running a project to create a reference translation dictionary to make the future translations as compatible as possible. Translation of IMP (web-based mail client) has been done and translation of KDE and GNOME are available. Webmin/Usermin translation and GNOME and KDE re-translation projects will be started in the second quarter of 2005 hopefully.

Changes acceptance by the global FOSS community

- In Foss world, testing and maintenance costs are often shared among the various users of the software.
- This is possible only if the changes made are accepted by the global community.
- Localization may involve changes in many different software components, each maintained by different project teams. Therefore, there should be a focused effort to ensure that all changes made are accepted by each of the teams, often by ensuring that the changes are made in a manner compatible with the future direction of the project team and so could be accepted by the project maintainers.
- The source codes at the main CVS which is maintained by maintainers are always used by Linux distributors.
- If you can't convince the project maintainer to accept your changes, your efforts will be at the risk being of isolated and hence will not be permanently accessible in GNU/Linux distributions.

A bit of Persian Linux localization Project history

- Feb. 2001: The feasibility study of localization of the GNU/Linux at ICT research center of Sharif University was started
- May. 2001: The proposal for “GNU/Linux localization project” was submitted
  - Some demos to show the strength of GNU/Linux against other alternatives such as Windows was prepared
  - Meetings with Government officials were held
    - Free software?= free of charge software
- After 2 years of hard work on Feb. 2003 the project was started
- May 2003: The Master Plan (management and executive plan) of the project was prepared

The short-term and long-term goals of the project

- Long term: A National Persian OS based on GNU/Linux (both client and server)
  - It takes a long time (at least 3 years).
  - Lack of Persian language support in GNU/Linux and Open Source software
  - Shortage of human resource
- Short term: providing the Persian language support in different levels of GNU/Linux: from Kernel to GUI
  - Persian specific libraries
  - General Persian support in GNU/Linux libraries
  - Application and GUI level support
- But the people and the Government can not wait for three years: A live/installable GNU/Linux CD shows the progress of the project and integrates the results

Completed sub-projects (2003-4)

- FriBiDi, shaping and joining
- Persian sorting, loose search, and locale
- Jalali Calendar
- Persian standard keyboard
- Persian OpenType reference font and rendering engine improvement
- A live Persian Linux CD based on Knoppix

Under progress sub-projects (2004-5)

- Persian Unicode support in
  - Glibc (persian numbers)
  - PostgreSQL
  - MySQL
  - IMP (web-based mail client)
  - Aspell (spell checker)
  - Mozilla suite analysis
  - Evolution (Personal Information Manager)
- Reference translation dictionary
- Persian GUI design regulations

Planned sub-project (2005)

- Persian Unicode support in
  - gtk+
  - Qt
  - ICU
  - Mozilla suite
- Openoffice.org Persian problem analysis
- GNOME re-translation
- KDE re-translation
- Webmin/Usermin translation
- Persian/English dictionary for Linux
- Font design

Free training content for all

- 120 hours of free GNU/Linux training materials (around 2000 slides)
- 120 hours of free E-learning GNU/Linux courseware
- 24 computer-based training GNU/Linux courses
- GNU/Linux courses for TV Educational Channel
- Financial support for GNU/Linux and Open Source books
- Financial support for GNU/Linux and Open Source projects at the Universities
- Iran GNU/Linux User Group (ILUG) Seminars and workshops

Cultural readiness for GNU/Linux

- Building business for IT companies: Teach them how to earn money from FOSS
- Government support for migration: creating the FOSS market
- Iran GNU/Linux User Group seminars
- GNU/Linux and Open Source promotion at universities, schools: Make the next generation ready for FOSS.
- TV programs and interviews
- Building success stories: pilot projects
- Promotion of governmental IT staff to use GNU/Linux and FOSS programs

References

- www.farsilinux.org
- http://www.iosn.net/foss/foss-general-primer/
- http://www.farsiweb.info/wiki/Standards/ISIRI6219
- projects.farsilinux.org
- http://www.isiri.org/std/2901.htm
- http://behdad.org/download/Publications/persiancomputing/slides_acro.pdf

Thank you for your attention.

Any question?

Contact:
Mohammad Khansari
Khansari@aictc.com
www.aictc.com