World Typography

Developments and Issues on the Free Desktop

ยินดีต้อนรับ

Unicode

typography

利用字型表现大爆炸的感觉

Edward H. Trager

Unifont.org

THE LINUX FOUNDATION

World Typography

THE LINUX FOUNDATION

World Typography

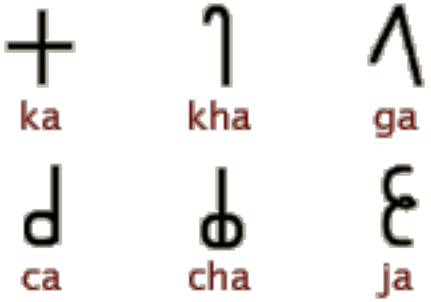
- Text Layout Features of Scripts
  - *Indic & Indic-derived Scripts*
  - *Middle East & African Scripts*
    - Arabic Script
  - *Asian Scripts*
  - *Multiple World Scripts*

THE LINUX FOUNDATION

World Typography

Text Layout Features of Scripts

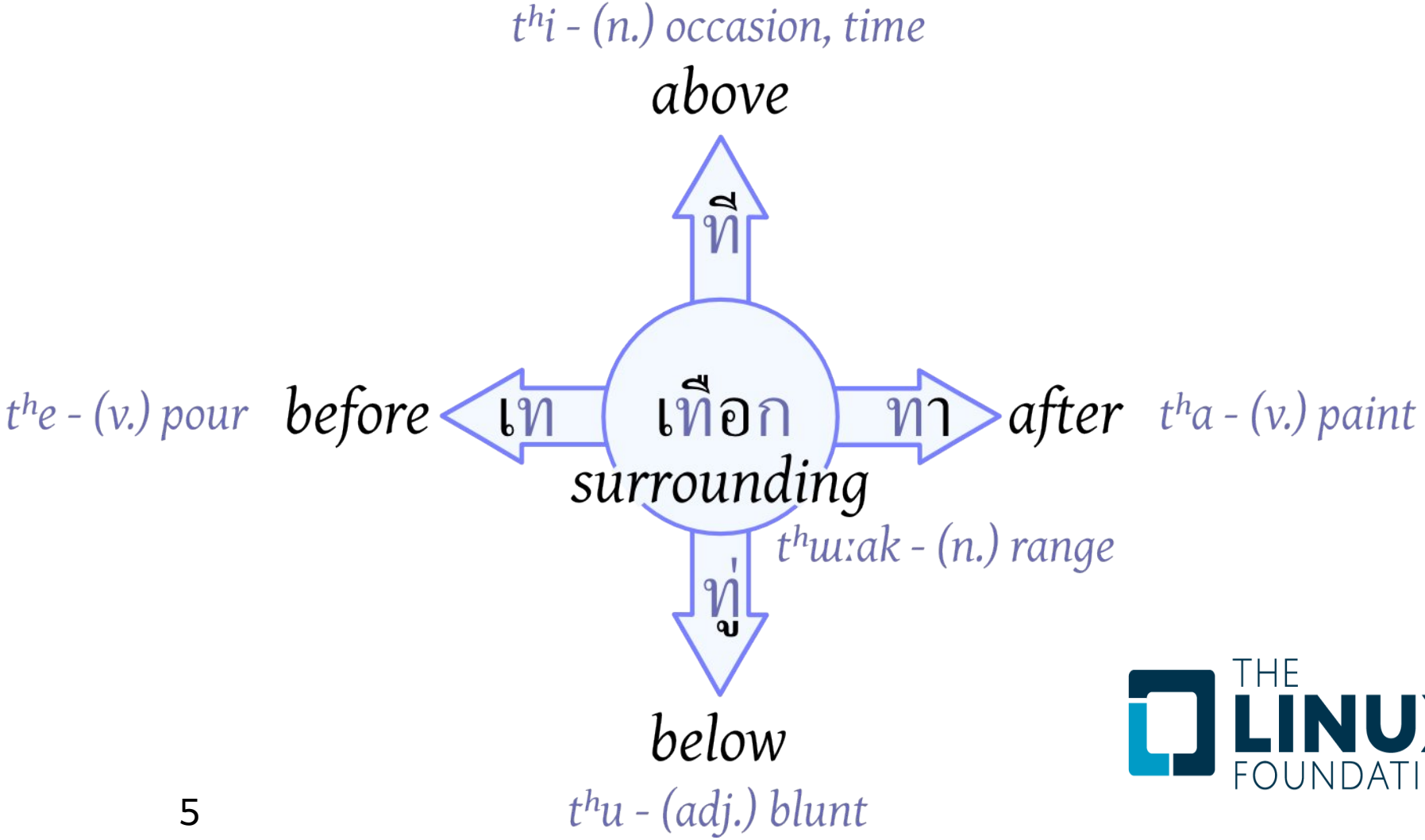
- *Indic & Indic-derived Scripts*
- *Middle East & African Scripts*
  - Arabic Script
- *Asian Scripts*
- *Multiple World Scripts*



THE LINUX FOUNDATION

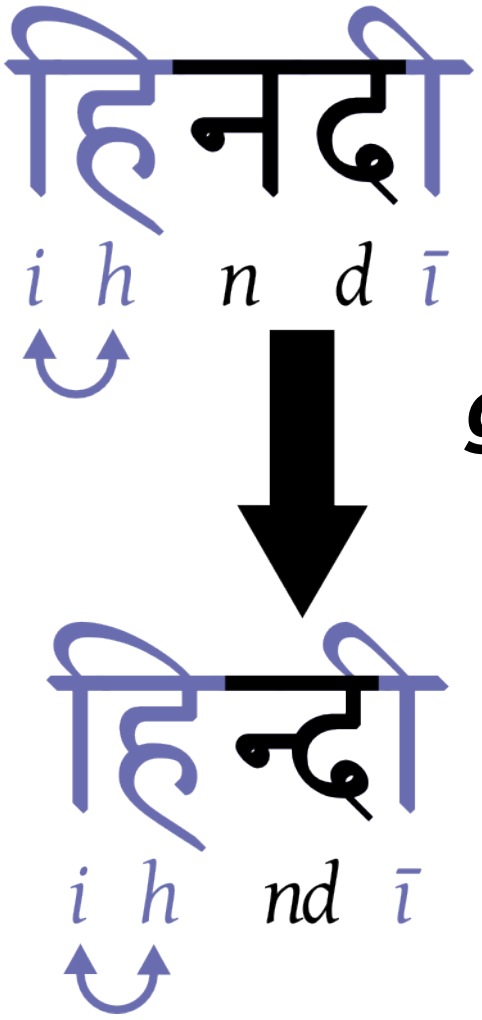
Indic & Indic-derived Scripts: Vowel Signs

Vowel positioning in Thai



Indic & Indic-derived Scripts: Consonant Clusters

"hindī"



Mandatory glyph transposition

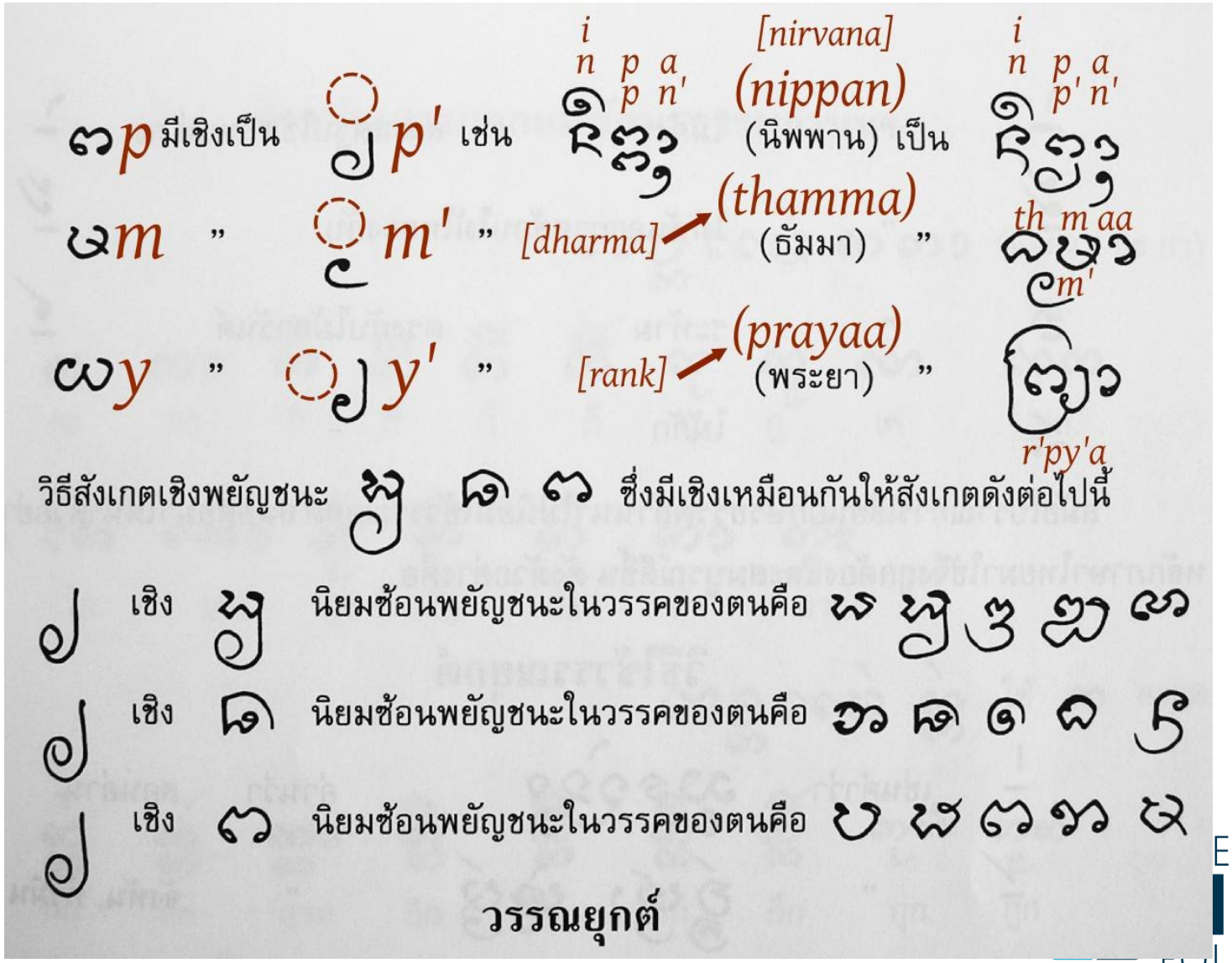
(vowel sign precedes consonant)

Mandatory glyph substitution

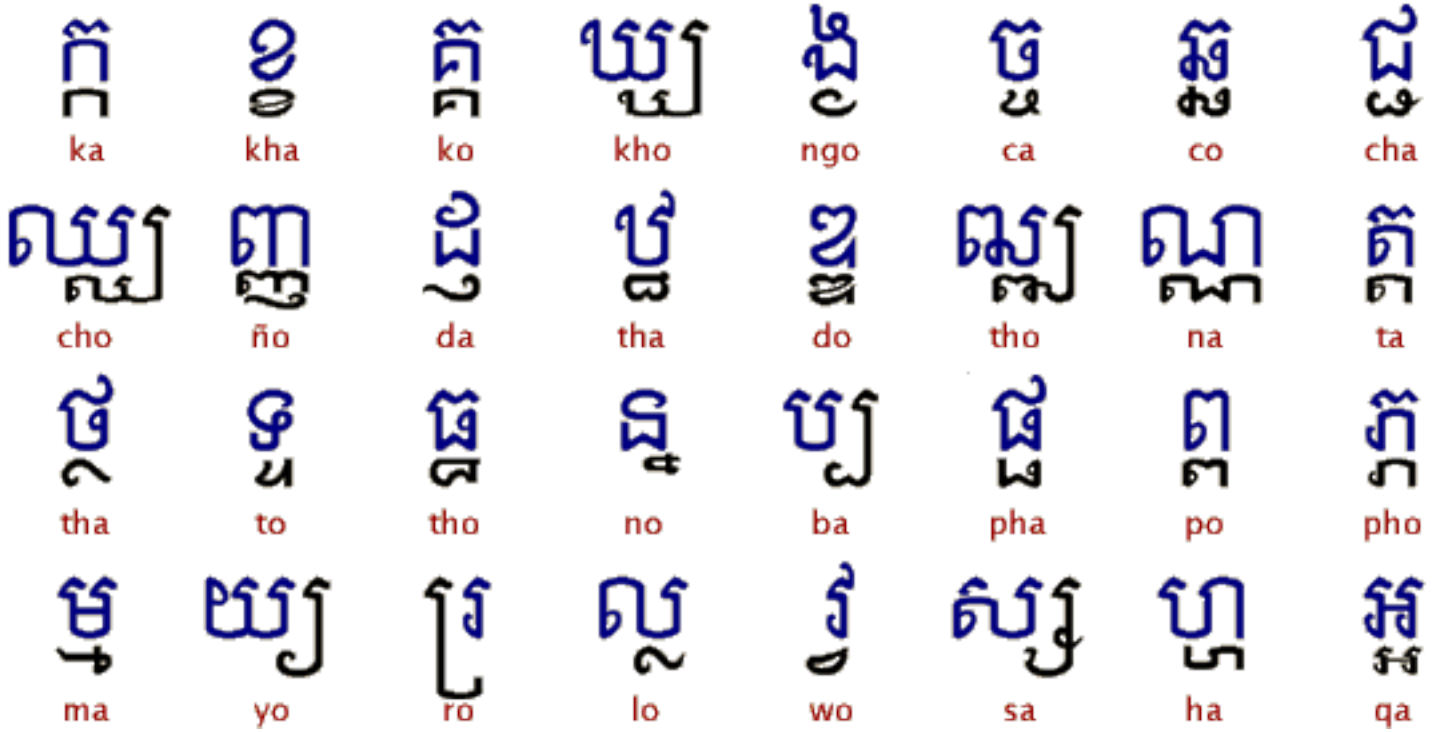
(half-form consonant conjunct)

THE LINUX FOUNDATION

Indic & Indic-derived Scripts: Vertically stacked consonants & clusters



Indic & Indic-derived Scripts: Vertically stacked consonants & clusters



(Image © http://www.ancientscripts.com/khmer.html)

Indic & Indic-derived Scripts: Spaceless Scripts

มนุษย์ทั้งหลายเกิดมามีอิสระ ... *Thai*

လူတိုင်းသည် တူညီ လွတ်လပ်သော *Myanmar*

มะนุดเกิดมามีเสธีพาบ ... *Lao*

Indic & Indic-derived Scripts: Spaceless Scripts

- **Thai**
- **Lao**
- **Khmer *(Cambodian)***
- **Myanmar *(Burmese)***
- **Lanna *(added to Unicode in 2006)***
- **Tai Dam *(Viet Tai, Unicode proposal exists)***
- **Tai Le *(Dehong Dai, now in Unicode)***
- **New Tai Le *(now in Unicode)***

Indic & Indic-derived Scripts: Spaceless Scripts

Allhumanbeingsarebornfreeandequal

English

Fussballweltmeisterschaftsqualifikationsspiel

German

THE LINUX FOUNDATION

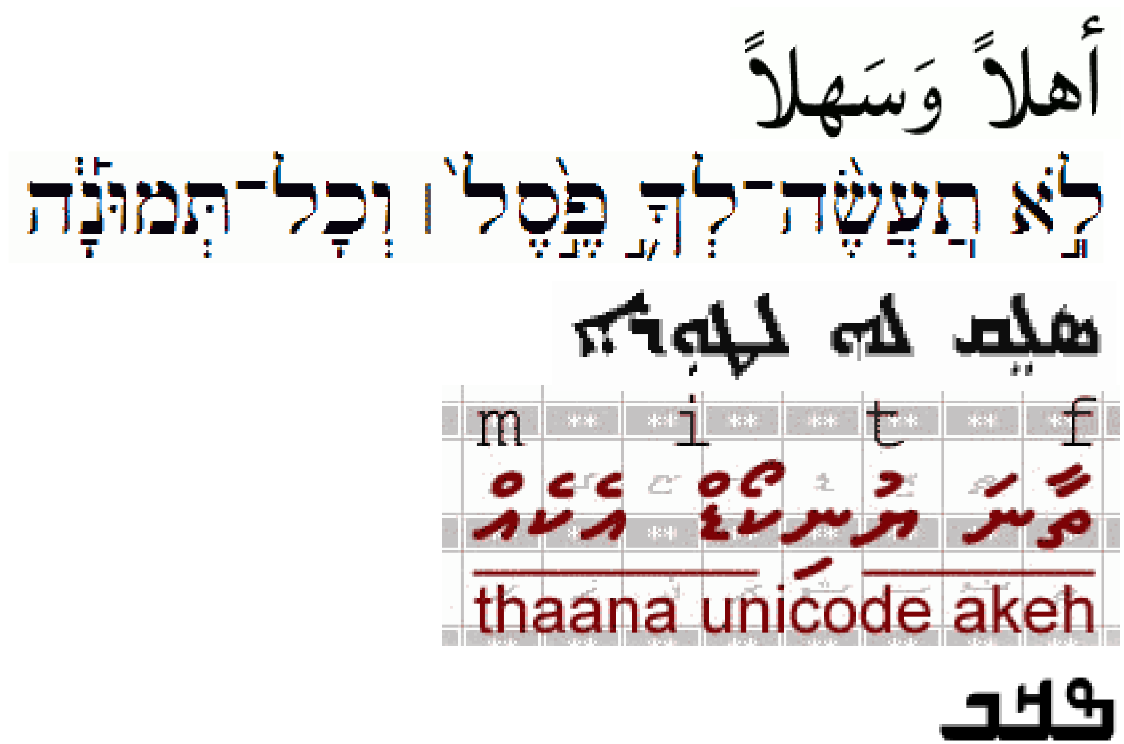
World Typography

- Text Layout Features of Scripts
  - *Indic & Indic-derived Scripts*
  - *Middle East & African Scripts*
    - Arabic Script
  - *Asian Scripts*
  - *Multiple World Scripts*

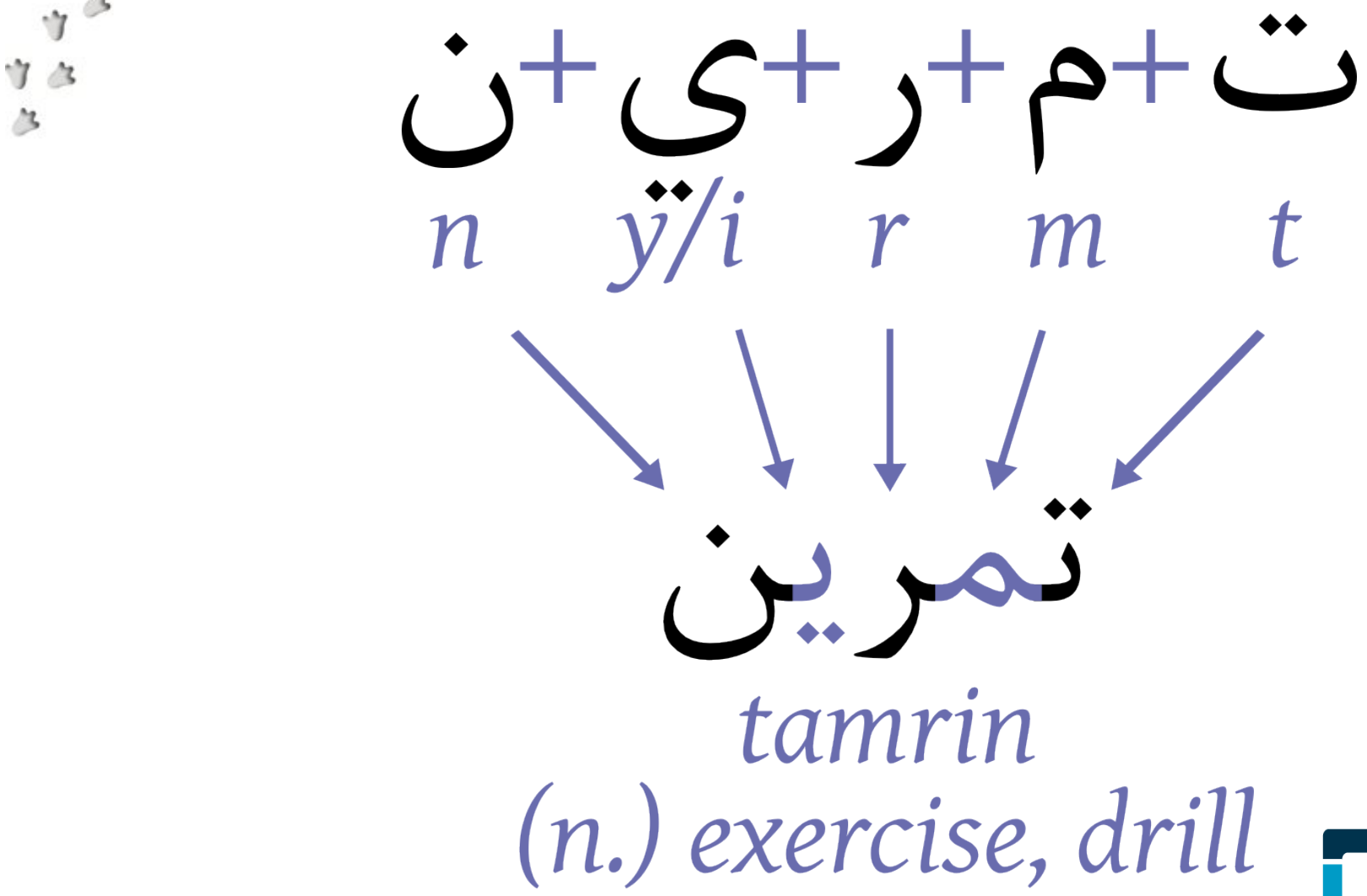
THE LINUX FOUNDATION

Middle East & African Scripts: Right-to-left Directionality

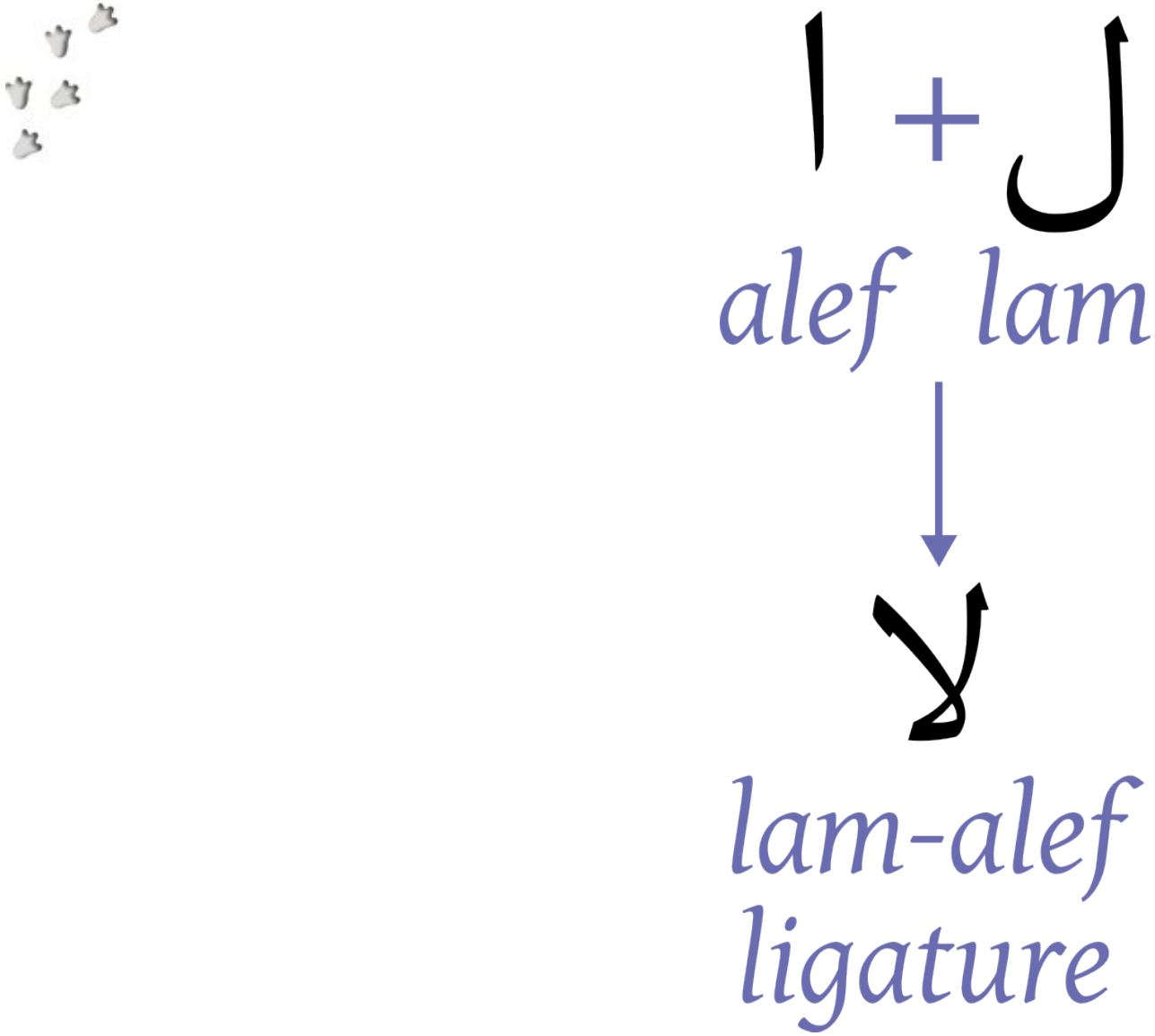
- **Arabic**
- **Hebrew**
- **Syriac**
- **Thaana**
- **N'Ko**



Arabic:
Positional Glyph Substitution



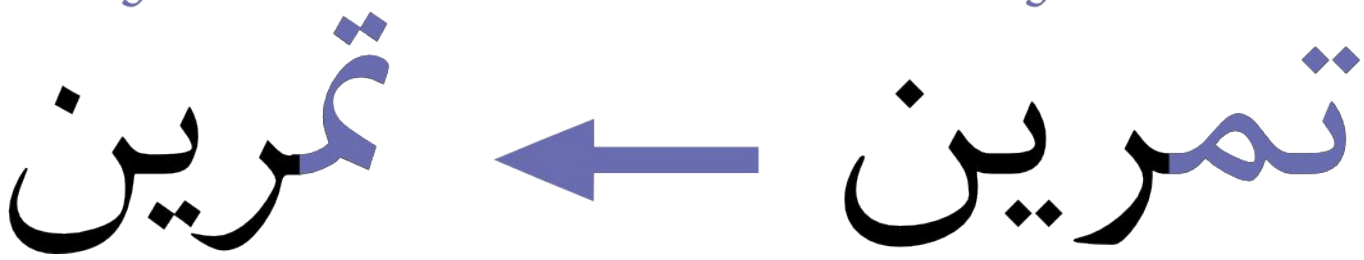
Arabic:
Mandatory Ligatures



Arabic:
Optional Stylistic Ligatures

Arabic OpenType font with set of extended ligatures

Arabic OpenType font without set of extended ligatures



THE LINUX FOUNDATION

Arabic: Kashida

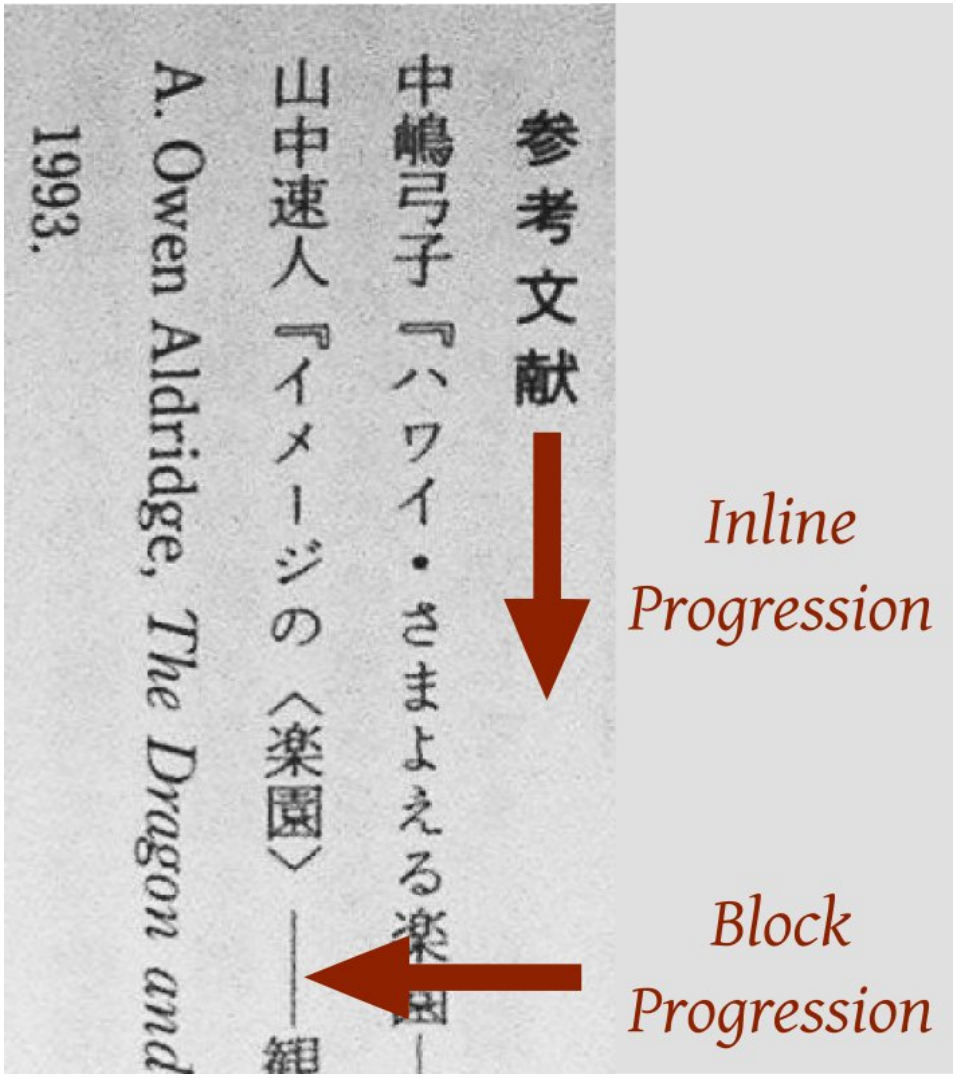
بســـــم الله الرحمن الرحيـــــم

World Typography

- Text Layout Features of Scripts
  - *Indic & Indic-derived Scripts*
  - *Middle East & African Scripts*
    - Arabic Script
  - *Asian Scripts*
  - *Multiple World Scripts*

利用TYPOGRAPHY表現大爆炸的感覺

Asian Scripts: Vertical Layout – Japanese & Chinese



THE LINUX FOUNDATION

Asian Scripts:
Vertical Layout – Traditional Mongolian

World Typography

Text Layout Features of Scripts

- *Indic & Indic-derived Scripts*
- *Middle East & African Scripts*
  - Arabic Script
- *Asian Scripts*
- *Multiple World Scripts*

THE LINUX FOUNDATION

Multiple World Scripts: Diacritical Marks

- **Diacritical Marks**
- **Vowel Marks**
- **Cantillation Marks**
- **Tone Marks**
- **etc. ...**

Multiple World Scripts: Baseline Adjustments

A. No baseline adjustment: inconsistent interline spacing

Beginning in the 1920s and 1930s, Marc Aurel Stein and others found thousands of scrolls in the Mogao Caves 莫高窟 near Dunhuang 敦煌. They included over 50 partial and complete Tao Te Ching 道德經 manuscripts. One written by the scribe So/Su Dan 素統 is dated 270 CE, and corresponds closely with the Heshang Gong 河上公 version. Another partial manuscript has the Xiang'er 想爾 commentary, which had previously been lost.

B. Baseline adjustment of Chinese: consistent interline spacing

Beginning in the 1920s and 1930s, Marc Aurel Stein and others found thousands of scrolls in the Mogao Caves 莫高窟 near Dunhuang 敦煌. They included over 50 partial and complete Tao Te Ching 道德經 manuscripts. One written by the scribe So/Su Dan 素統 is dated 270 CE, and corresponds closely with the Heshang Gong 河上公 version. Another partial manuscript has the Xiang'er 想爾 commentary, which had previously been lost.

THE LINUX FOUNDATION

World Typography: Text Layout Requirements Summary

- **Glyph Substitution**
- **Glyph Positioning**
- **Baseline Adjustments**
- **Vertical Metrics**
- **Stylistic Alternates**
- **Optional Ligatures**
- **Swash Forms**

THE LINUX FOUNDATION

Free Desktop Scorecard: Problems & Solutions

THE LINUX FOUNDATION

The Free Desktop Scorecard

- Problems & Solutions
  - *Script Rendering*
  - *Editing Text*
  - *Font Problems*
    - Pan Unicode fonts
    - Font Substitution issues
  - *Input Methods*

THE LINUX FOUNDATION

The Free Desktop Scorecard

Problems & Solutions

- *Script Rendering*
- *Editing Text*
- *Font Problems*
  - Pan Unicode fonts
  - Font Substitution issues
- *Input Methods*

THE LINUX FOUNDATION

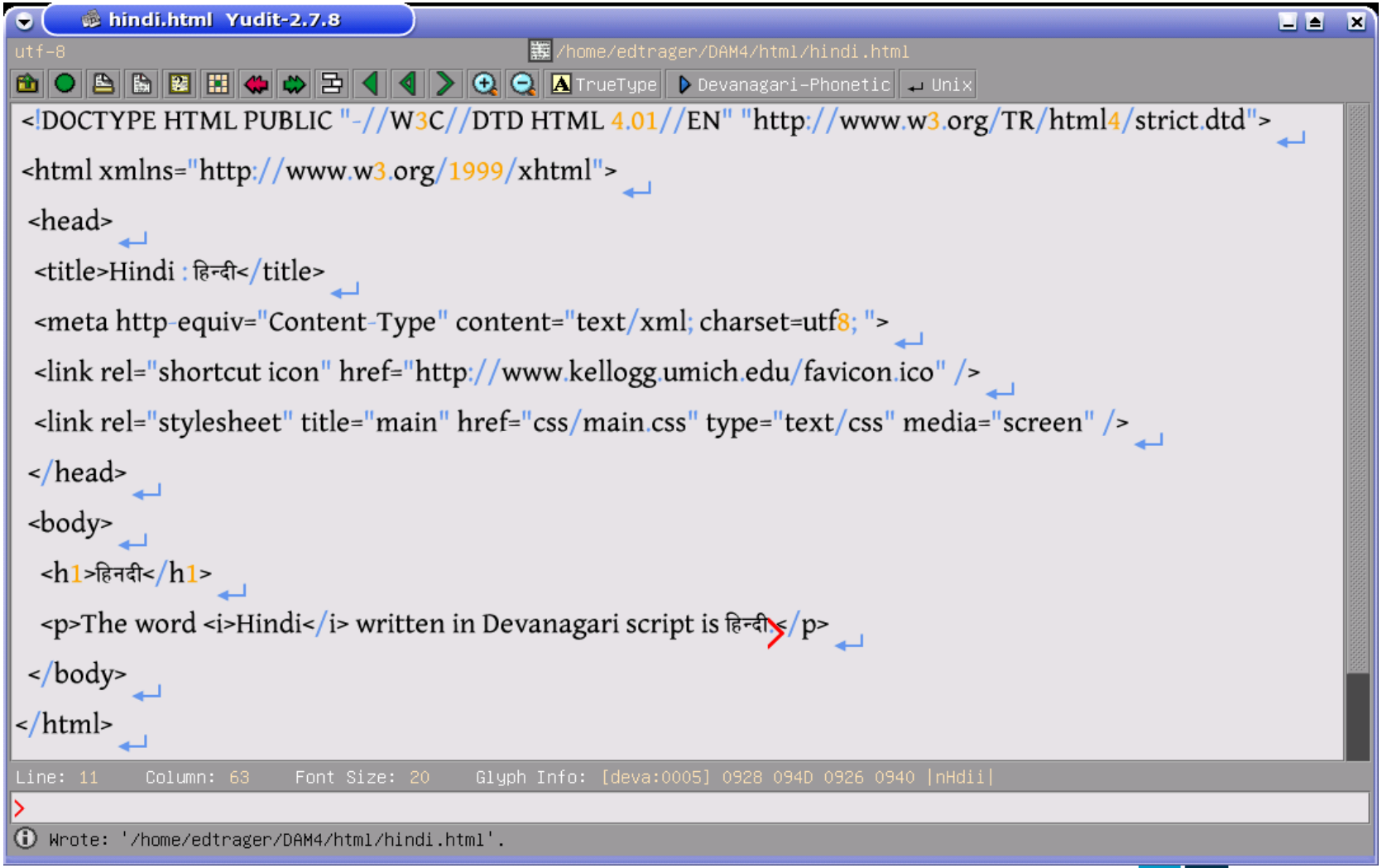
Script Rendering Support : Problems

- **Inconsistent Support – different layout engines**
- **Buggy Support – esp. Indic**
- **Inefficient Support**
- **Not Supported At All – Myanmar, Mongolian ...**

THE LINUX FOUNDATION

Inconsistent Script Support

Devagari in Yudit : Good

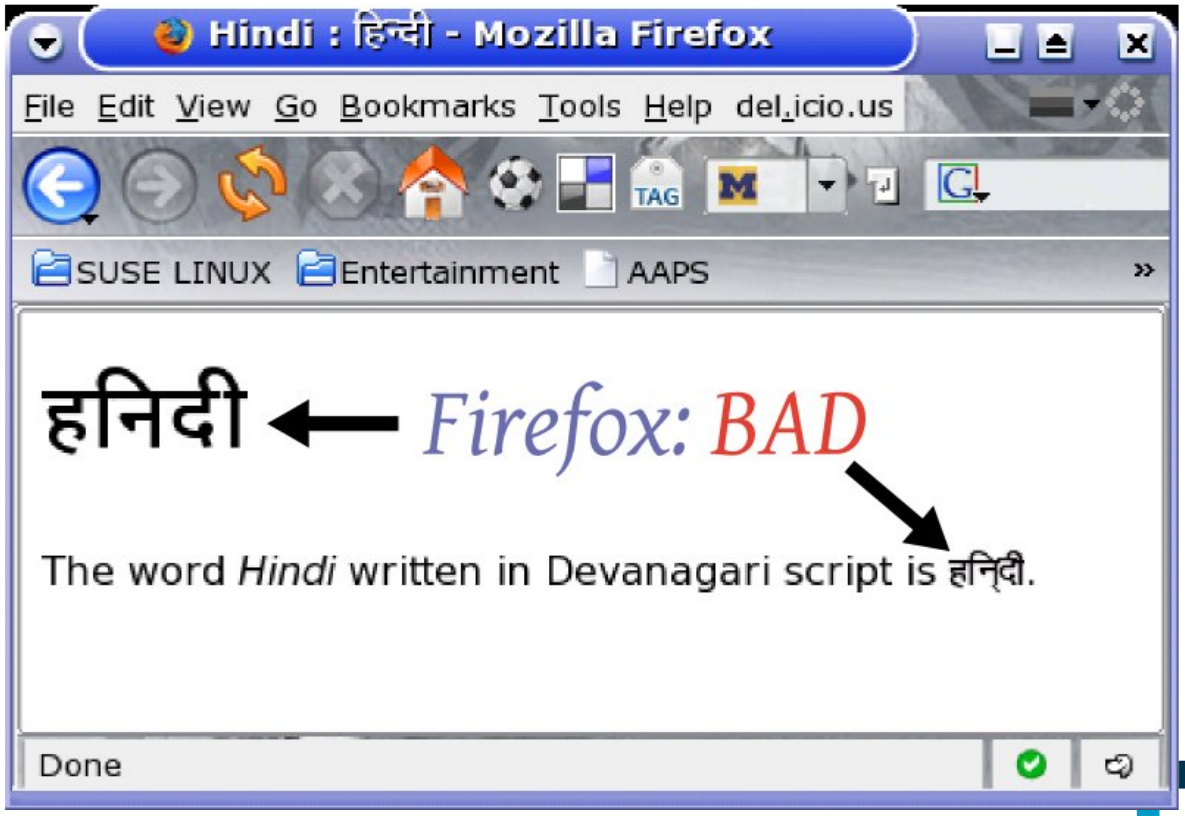


UX FOUNDATION

Inconsistent Script Support

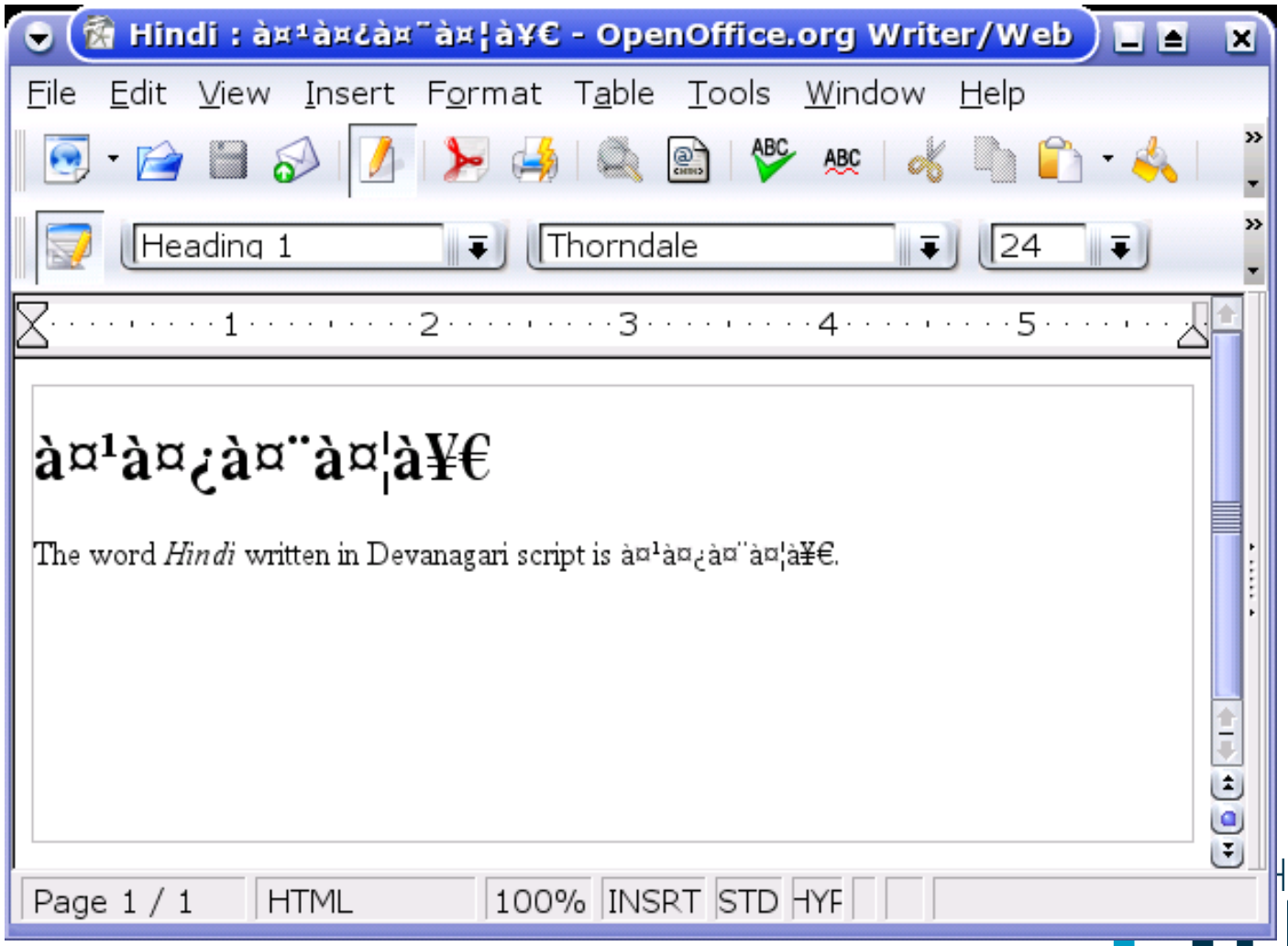
Devagari in Firefox : BAD

KDE Window manager: ugly but correct



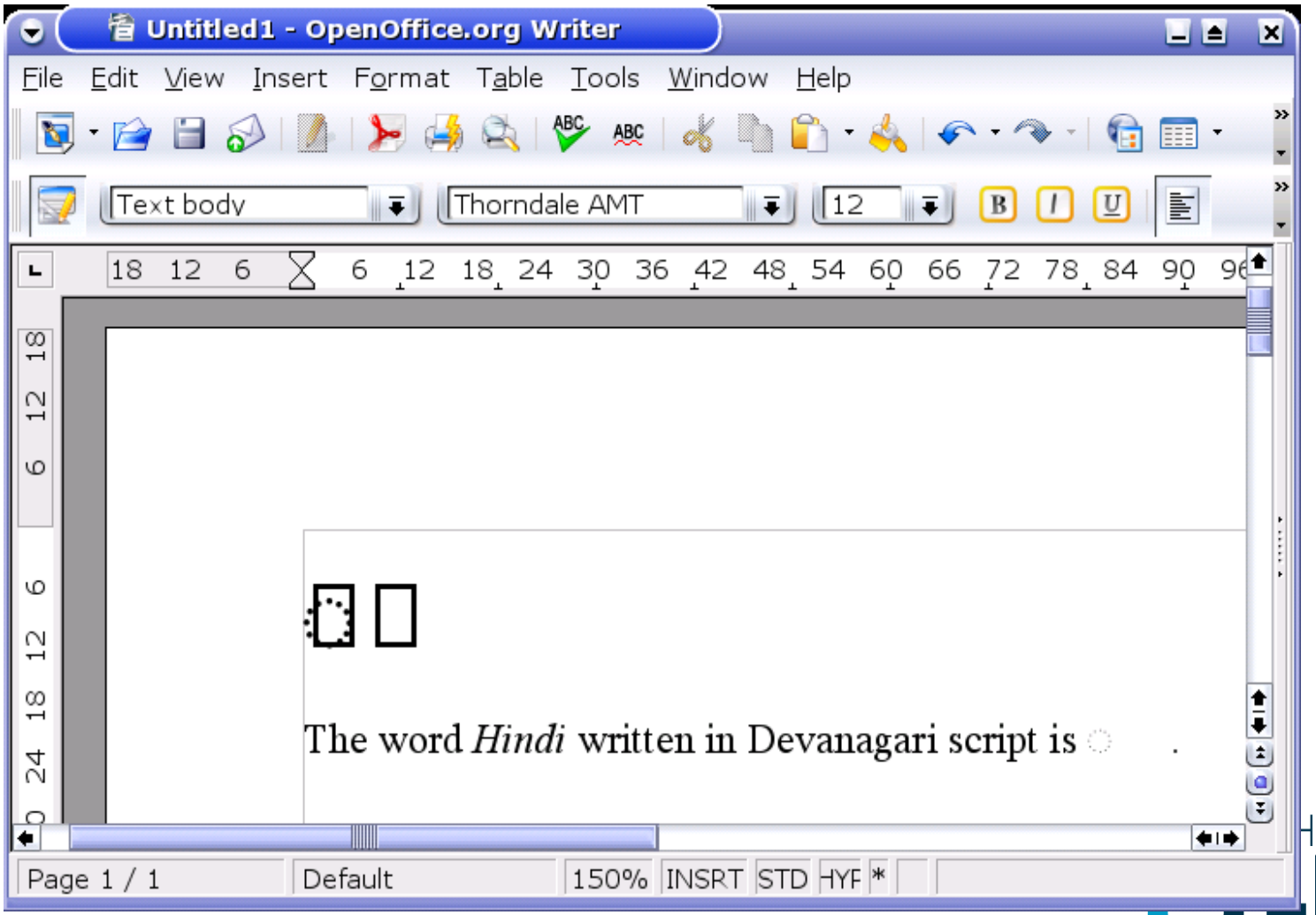
Inconsistent Script Support

Devagari in OO Web : BAD – UTF-8 recognition



Inconsistent Script Support:

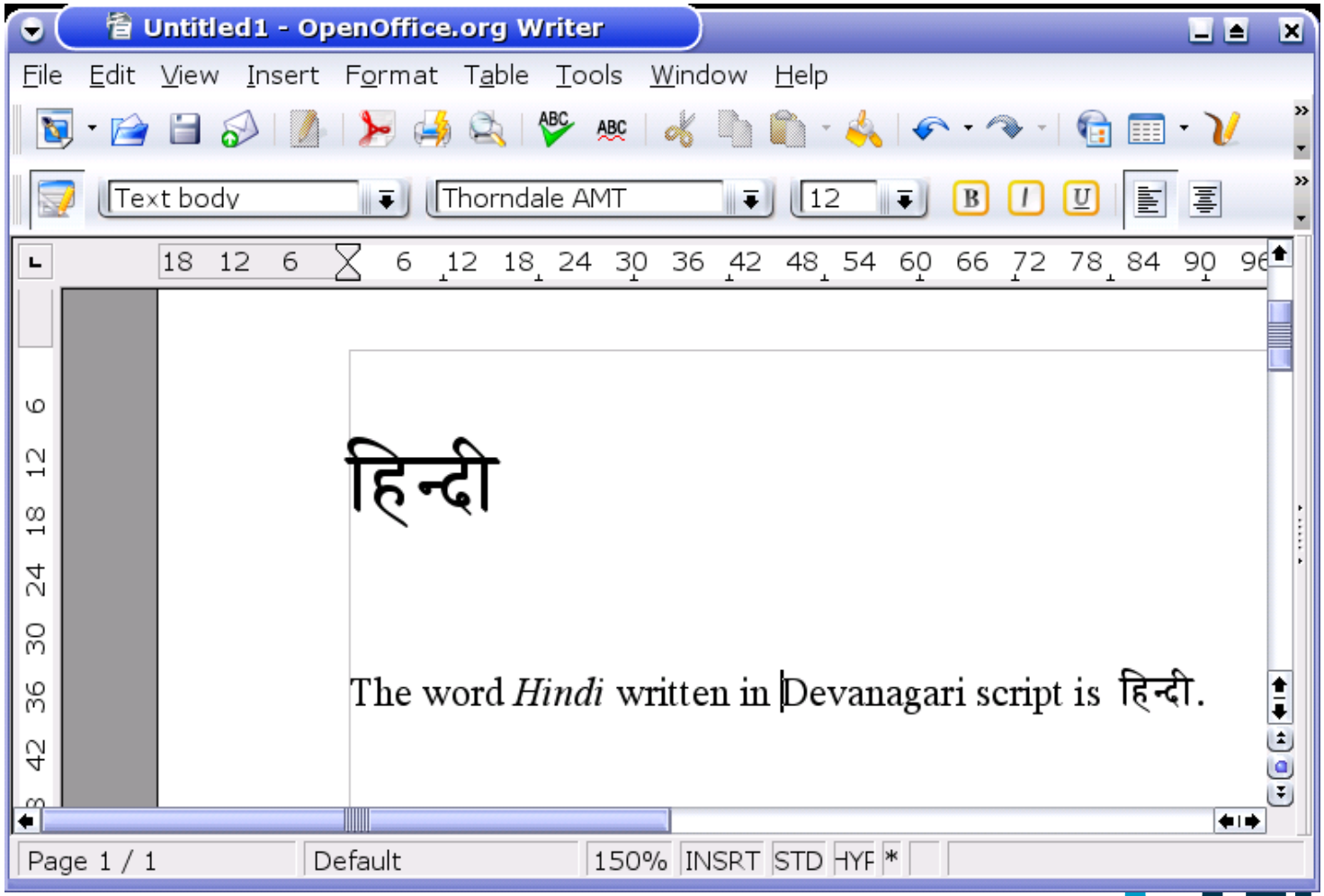
Devagari in OO Writer : BAD font substitution



Inconsistent Script Support

Devagari in OO Writer :

OK (After manually selecting Chandas font)



Buggy Script Support

- *On May 24, 2007, Behdad Esfahbod wrote on the HarfBuzz mailing list:*

"Note that the previous consensus that Qt's shapers are preferred was about the Indic shaper, not about them all in general. For Indic shaper, nobody knows which one (Pango, Qt, ICU) is the best. What we know is that maintainers of Pango and ICU are not happy about theirs. That's how it was decided to use Qt's. But at the end it really doesn't matter as the three are mostly the same, and each has its own share of bugs."

Inefficient Script Support

- *On October 3, 2006, Trager to Esfahbod:*
- *Trager:* "In looking at the (Arabic shaper) code, I could not help noticing that first the UTF-8 string is converted to a UCS4 string ... but then in the subsequent for(;;) loop, the UTF-8 string is converted a second time, but this time character-by-character ..."
- *Esfahbod:* "Your observations are all correct."
- Ubuntu Bug #32561: "Pango-enabled Firefox is much slower"

Scripts Not Supported At All

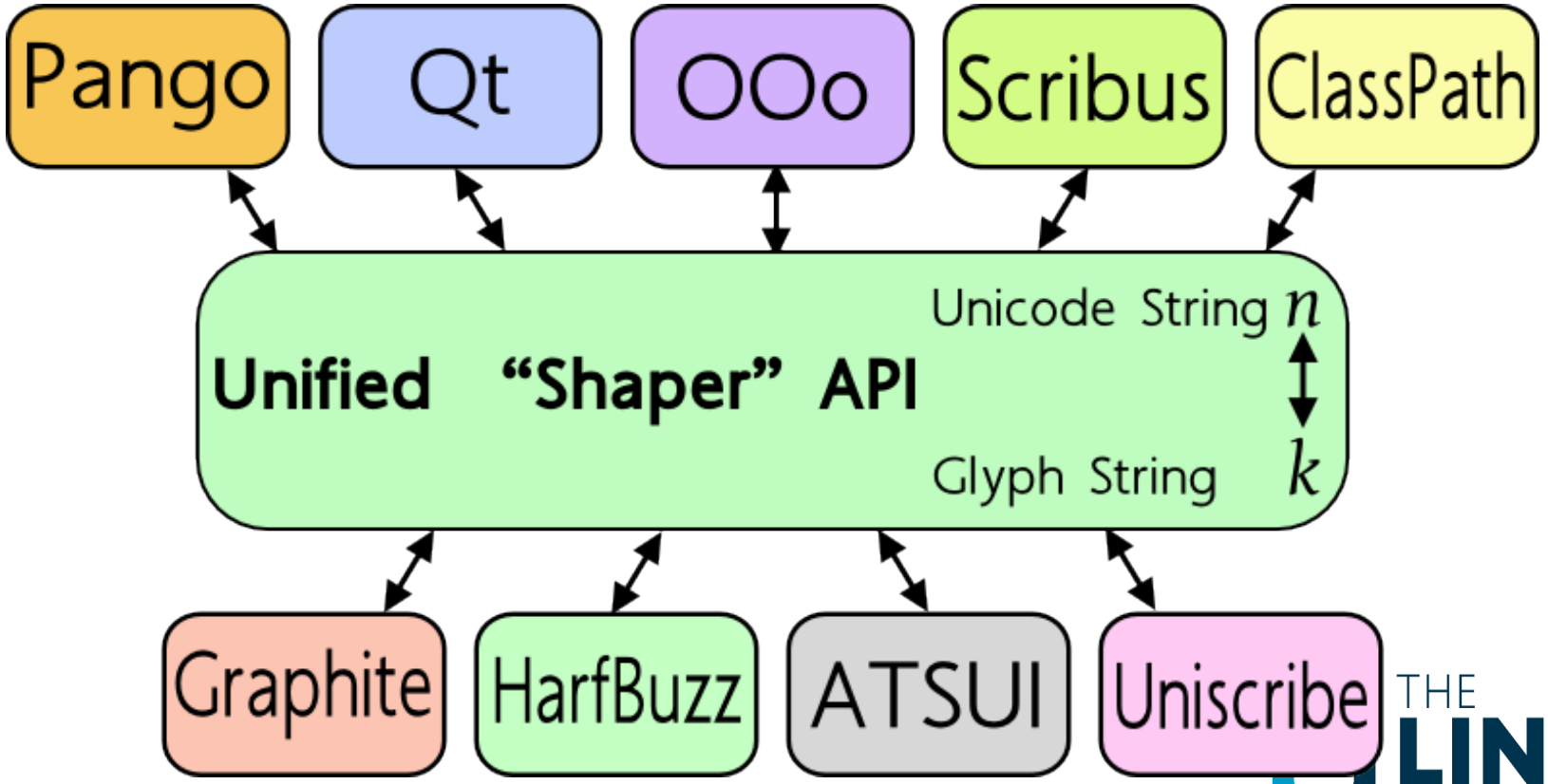
- **Myanmar : 32.3 million speakers**
- **Classical Mongolian : 5 million speakers**
- **Lanna : 6 million speakers**
- ***etc*. ...**

Compare to :

- ***Modern Greek : 12.2 million speakers***
- ***Georgian: 4.1 million speakers***

Script Rendering Support : Solution

- Text Layout Summit @ Boston Gnome Live! 2006
- Agreement on unified “shaper” API
- “Low hanging fruit” is HarfBuzz OpenType Library



Script Rendering : Word breaking & Syllabification of Spaceless Scripts

- **Problems:**
  - Software only available for Thai (libThai)
  - Little or no support for Khmer, Myanmar, Lao and others
- **Solutions:**
  - Create an object-oriented framework for rule- & dictionary-based word-boundary analysis
  - Plug into unified text rendering pipeline

มนุษย์ทั้งหลายเกิดมามีอิสระ ... *Thai*

လူတိုင်းသည် တူညီ လွတ်လပ်သော *Myanmar*

ມະນຸດເກີດມາມີເສຣີພາບ ... *Lao*

The Free Desktop Scorecard

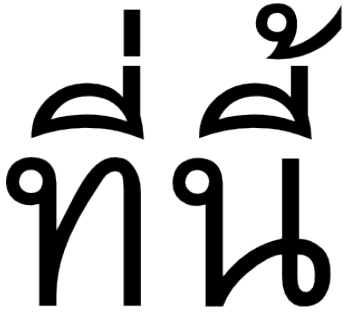
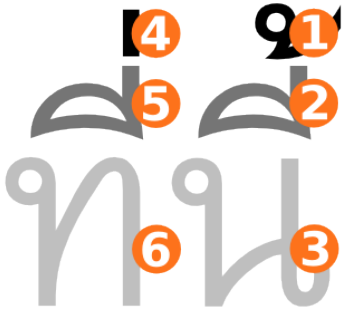
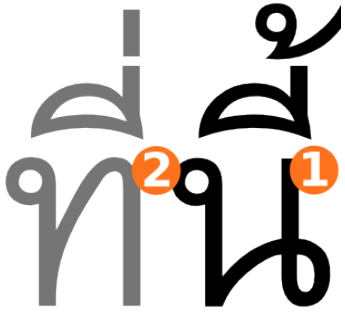
- Problems & Solutions
  - *Script Rendering*
  - *Editing Text*
  - *Font Problems*
    - Pan Unicode fonts
    - Font Substitution issues
  - *Input Methods*

THE LINUX FOUNDATION

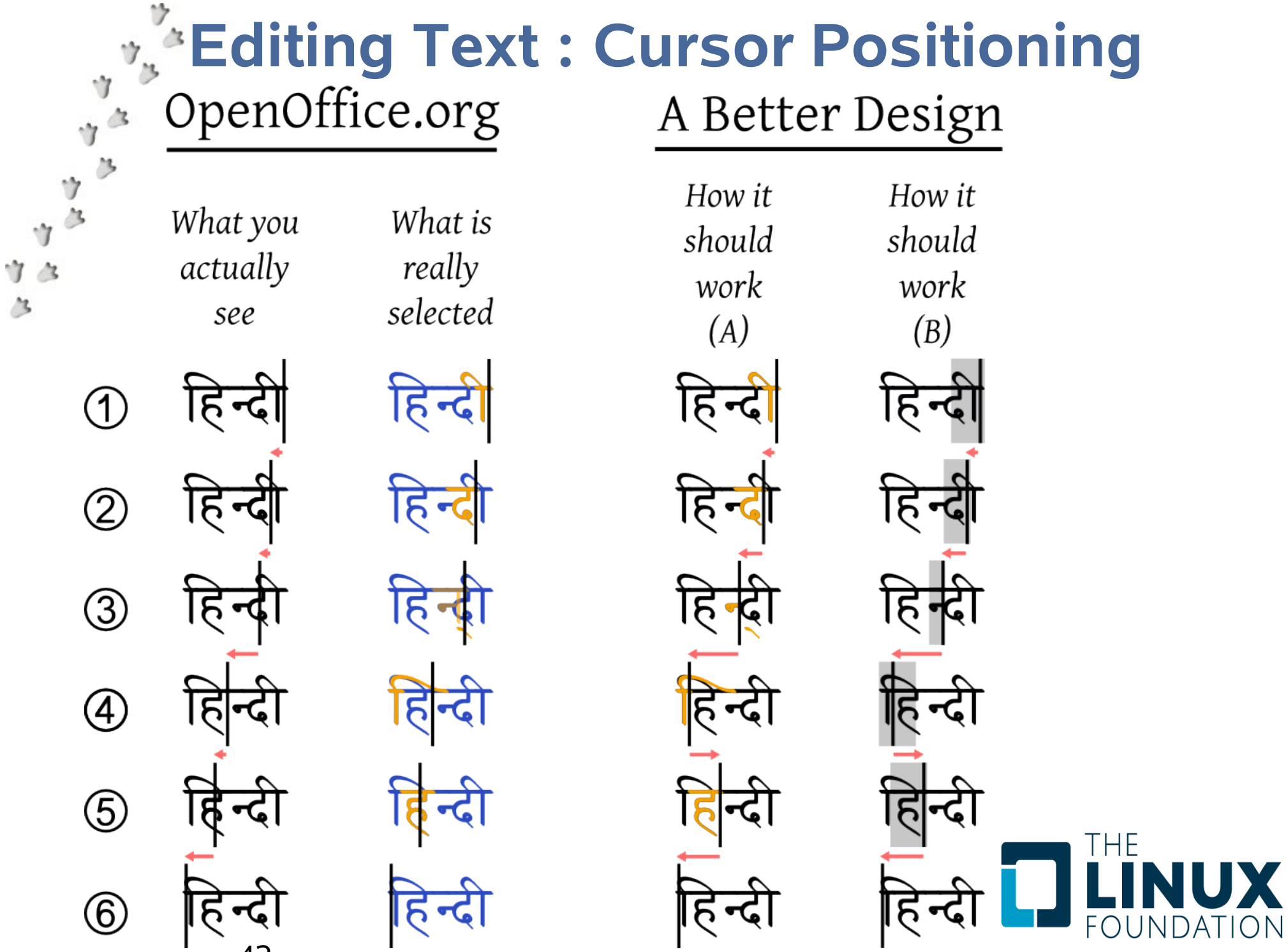
Editing Text : Problems

- **Backspace or delete gobbles entire cluster**
- **Cursor positioning makes no sense**

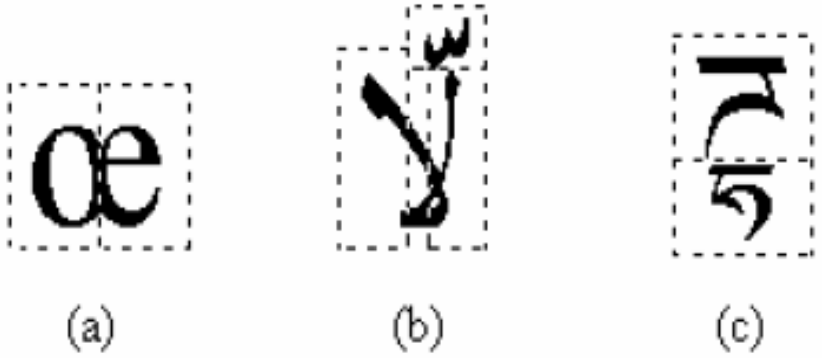
Editing Text : Delete Gobbles Cluster

Target text	Typing order	Erasing order	Erasing order
			
		OOo KWord Gimp	Inkscape Yudit

Editing Text : Cursor Positioning



Editing Text : Solutions



- **Ligatures are not monolithic**
- **Underlying characters are accessible**
- **Access rectangular areas corresponding to glyph components**
- **Highlighting & mouse selection handled by text layout engine (Graphite, HarfBuzz, etc)**
- **No complex programming required**

The Free Desktop Scorecard

Problems & Solutions

- *Script Rendering*
- *Editing Text*
- *Font Problems*
  - Pan Unicode fonts
  - Font Substitution issues
- *Input Methods*

THE LINUX FOUNDATION

Font Problems : Pan-Unicode Fonts

- **We don't need Pan-Unicode fonts because:**
  - An increasing number of high-quality fonts for specific scripts are being produced by dedicated groups
  - Flexibility of fontconfig allows construction of virtual fonts that cover Unicode code space better
- **But groups persist in adding more Unicode blocks to fonts:**
  - Many users were not happy when Arabic first added to DejaVu
- **Asian CJK fonts also often have inferior Latin/Greek/Cyrillic (LGC) glyphs**

THE LINUX FOUNDATION

Font Solutions

- **Use specific fonts for specific orthographies**
- **Extend Fontconfig syntax to be able to specify fonts by script and orthography directly**

Font Solutions

Example: Vietnamese expect more vertical spacing:

DejaVu Sans

Đời cha ăn mặn,
đời con khát nước. 7pt.
Đời cha ăn mặn,
đời con khát nước. 8 pt.
Đời cha ăn mặn,
đời con khát nước. 10 pt.
Đời cha ăn mặn,
đời con khát nước. 12 pt.

VU Pho Tho

Đời cha ăn mặn,
đời con khát nước. 7 pt.
Đời cha ăn mặn,
đời con khát nước. 8pt.
Đời cha ăn mặn,
đời con khát nước. 10 pt.
Đời cha ăn mặn,
đời con khát nước. 12 pt.

THE LINUX FOUNDATION

Font Solutions

... so extend fontconfig syntax to specify fonts by script and orthography directly:

```
<script>
  <name>latn</name>
  <orthography>
    <name>vn</name>
    <alias>
      <family>VU Pho Tho</family>
      <family>DejaVu Sans</family>
      <default><family>sans</family></default>
    </alias>
  </orthography>
</script>
```

Font Solutions

Japanese & Chinese prefer different fonts:

直骨人 AR PL ShanHeiSun Uni 中

直骨人 AR PL New Sung 中

直骨人 cwTeXMing 中

直骨人 Sazanami Mincho 日

直骨人 IPAPGothic 日

直骨人 Konatu 日

Font Solutions

Extended XML syntax would solve the problem:

```
<script>
  <name>hani</name>
  <orthography>
    <name>zh</name>
    <alias>
      <family>AR PL ShanHeiSun Uni</family>
      <default><family>serif</family></default>
    </alias>
  </orthography>
  <orthography>
    <name>ja</name>
    <alias>
      <family>Sazanami Mincho</family>
      <default><family>serif</family></default>
    </alias>
  </orthography>
</script>
```

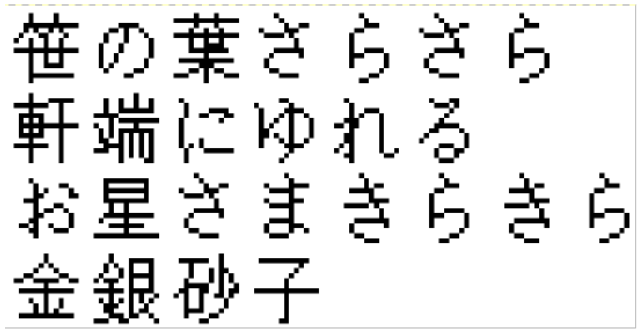
Configuring Fonts for Screen & Print

- **Problems:**
  - Fontconfig only defines serif, sans, monospace categories
  - Appropriate only for Western typography
- **What additional categories would be most useful?**
- **Solutions:**
  - ***Screen*** category
  - ***Print*** category

THE LINUX FOUNDATION

Configuring Fonts for Screen & Print

Screen - small to medium pixel sizes:
GNU Unifont or Wen Quan Yi bitmap font



Screen - larger pixel sizes: IPA Gothic

笹の葉さらさら
軒端にゆれる
お星さまきらきら
金銀砂子

Print: IPA Mincho

笹の葉さらさら
軒端にゆれる
お星さまきらきら
金銀砂子

Font Solutions : Fostering an Open Typography Community

- **Problems:**
  - "Free font" typographers historically worked independently, little cooperation
  - Often not aware of licensing issues
  - Often used unclear, ad-hoc, or incompatible licenses
- **Solution:**
  - **SIL's Open Font License** encourages an Open typographic community to spring up and grow.

THE LINUX FOUNDATION

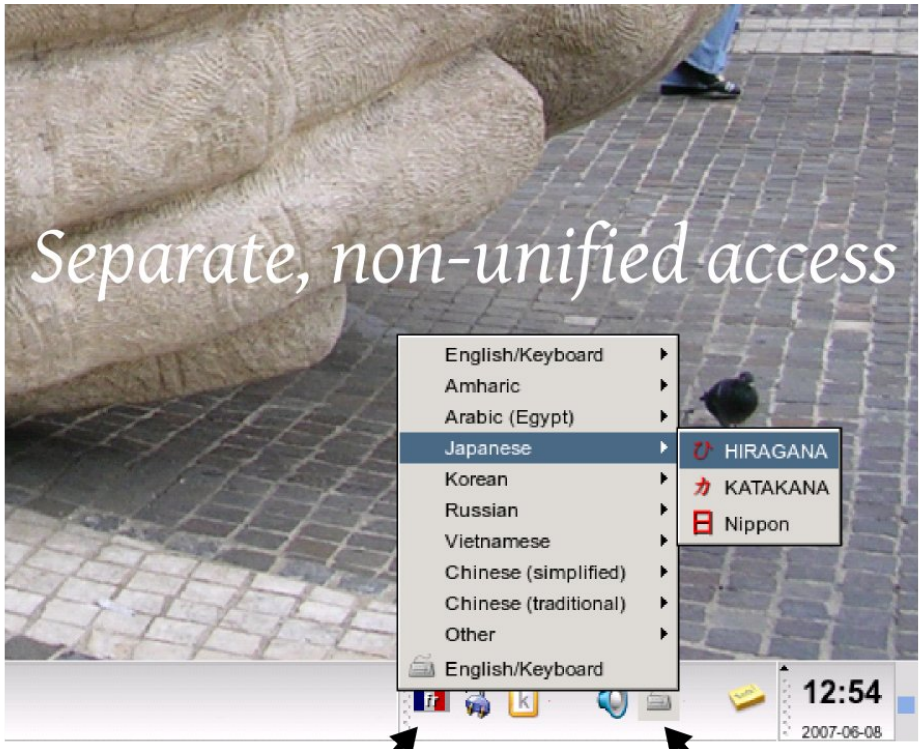
The Free Desktop Scorecard

Problems & Solutions

- *Script Rendering*
- *Editing Text*
- *Font Problems*
  - Pan Unicode fonts
  - Font Substitution issues
- *Input Methods*

Input Methods

- **Problem:**
  - Keyboard Layouts & Input Methods not unified



KDE Keyboard tray access

SCIM IME tray access

Input Methods

- **Solutions:**
  - Streamline & unify access to keyboard layouts and IMEs on the desktop

Wrap Up
Opportunities for the FLOSS community in the development pipeline

- **2nd Annual Text Layout Summit @ aKademy in Glasgow July 4-6, 2007**
- **Make unified text rendering pipeline a reality**
- **Increasing collaboration with non-traditional partners is already proving very fruitful**
- **Have a roadmap to implement new scripts as they are approved in Unicode**
- **Take a more pro-active role in pushing the Unicode Standard & World Typography on the Free Desktop to the next milestone.**

THE LINUX FOUNDATION

Resources

- **An Introduction to Indic Scripts by Richard Ishida (2003)** http://people.w3.org/rishida/scripts/indic-overview/
- **Free Desktop Text Layout Working Group** http://freedesktop.org/wiki/TextLayout
- **International Text Layout & Typography: The Big And Future Picture by Ed Trager** http://www.unifont.org/textlayout/TheBigPicture.pdf
- **HarfBuzz Mailing List** http://lists.freedesktop.org/mailman/listinfo/harfbuzz
- **Open Font License** http://scripts.sil.org/OFL
- **Unifont.org** http://www.unifont.org

THE LINUX FOUNDATION

World Typography

Developments and Issues on the Free Desktop

ยินดีต้อนรับ Unicode typography

利用字型表现大爆炸的感觉

Edward H. Trager
Unifont.org

THE LINUX FOUNDATION

The following talk provides an in-depth review of developments and issues related to world typography – that is to say internationalized text services (text layout, text input methods, fonts and font management issues) on the Free Desktop

World Typography

Let us start by looking at the requirements for world typography.

Last year at the **Gnome Live! Text Layout Summit** in Boston, I talked about "i*nternational"* text layout and typography.

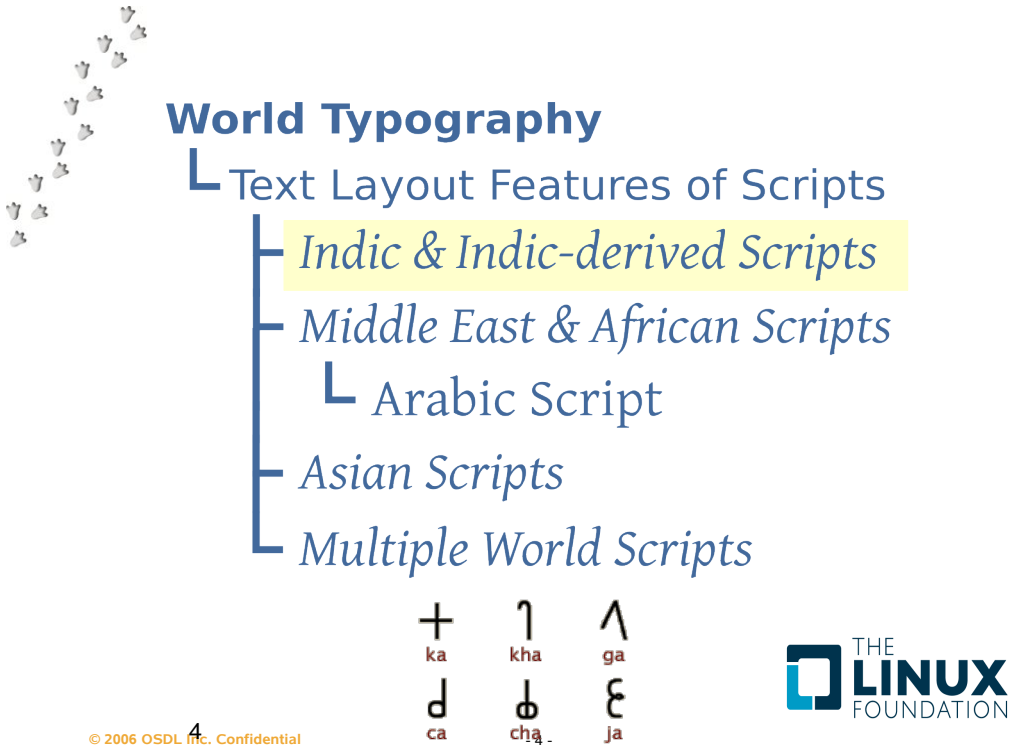
The world "international" emphasizes the differences and divisions between nations and states. So this year I've decided to use the word "*world*" instead in order to emphasize the oneness of our collective efforts to create a Free Desktop usable by all people everywhere in the world.

World Typography

- Text Layout Features of Scripts
  - *Indic & Indic-derived Scripts*
  - *Middle East & African Scripts*
    - Arabic Script
  - *Asian Scripts*
  - *Multiple World Scripts*

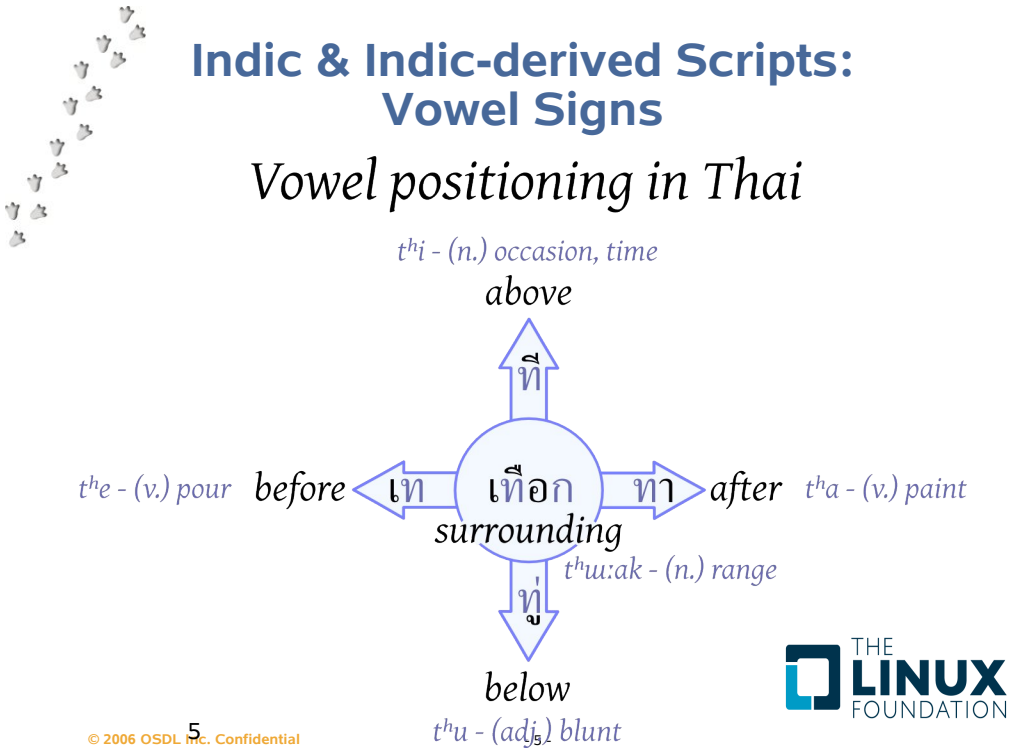
THE LINUX FOUNDATION

To insure that we all have a common understanding of the problem domain, let us start with a general review of the text layout features of world scripts.



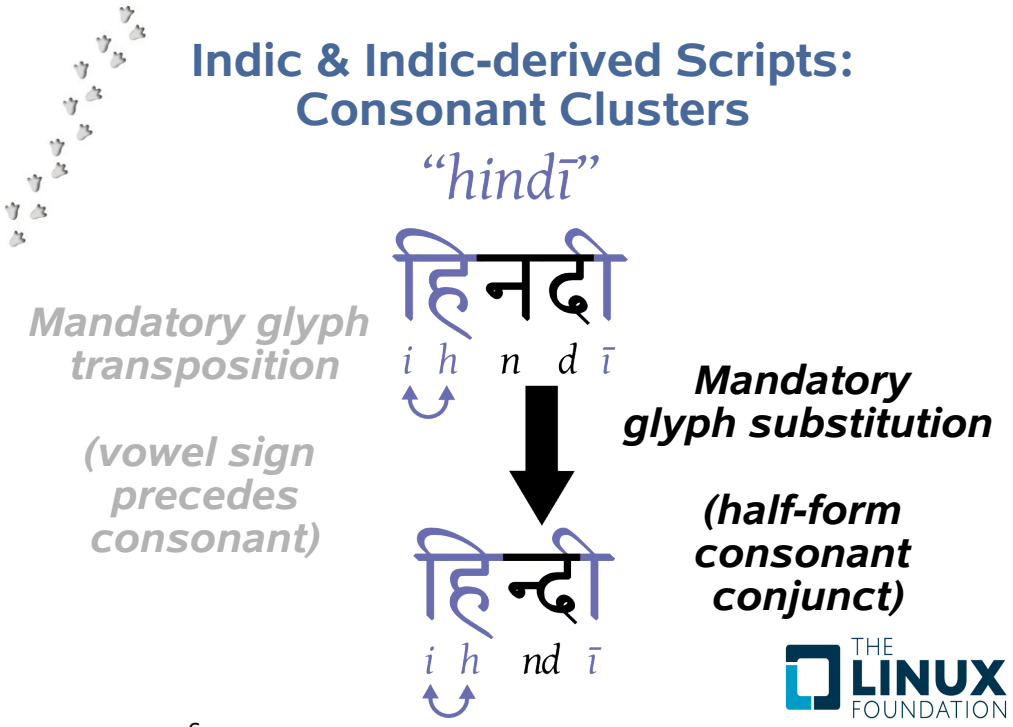
Let's start with the Indic- and Indic-derived scripts of South and Southeast Asia. All of these scripts ultimately derive from the Brahmi script which originated by some estimates even earlier than the 5th century BCE.

Most of these scripts are *abugidas* which consist of consonants with an inherint vowel sound. Additional vowel signs are used to change the inherint vowel sound.



A common theme in many of these scripts is that the vowel signs may appear before, after, above, below, or even as a cluster of signs surrounding a base consonant or cluster of consonants.

Thai is used in the illustration shown here.



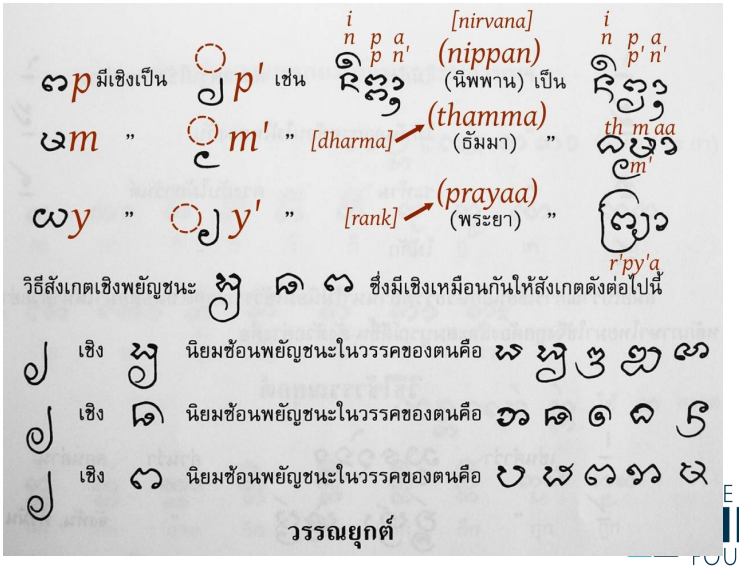
Consonant clusters (or consonant conjuncts) are another very common feature in the Indic and Indic-derived scripts.

In the Devanagari example shown here, the “n” and “d” letters in the word “Hindi” join horizontally to form a “half-form” consonant conjunct. In this simple case, the vertical line which historically relates to the inherint vowel “a” in the “na” letter is dropped so that the half-form “n” can join to the following letter “da”.

About 60% of the conjuncts in Devanagari are formed this way. However, sometimes two consonant glyphs join vertically instead of horizontally. In other cases, some simplification of one or more of the components takes place. And in some cases the conjunct form is completely unrecognizable and does not appear to have been derived from the component glyph forms at all.

About a thousand conjunct consonants are in common use in Devanagari. Most of these are combinations of two or three consonants. However four-consonant conjuncts exist, and at least one conjunct with five consonants is well-known.

Indic & Indic-derived Scripts: Vertically stacked consonants & clusters

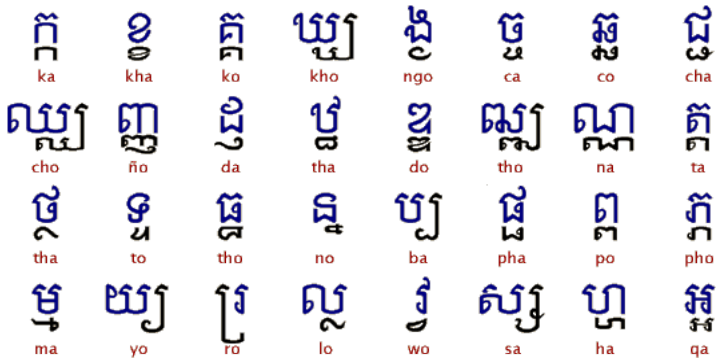


Vertically stacked consonants and consonant clusters are quite common in some Indic and Indic-derived scripts. Here some examples from the Lanna script of Northern Thailand are shown.

Note how the subjoined consonant forms shown are simpler and visually distinct from the regular consonant forms.

The Lanna script was approved for inclusion in Unicode in 2006.

Indic & Indic-derived Scripts: Vertically stacked consonants & clusters



(Image © http://www.ancientscripts.com/khmer.html)

Other Indic-derived scripts with similar features include the Khmer and Tibetan scripts. Some subjoined Khmer consonant forms are shown here along with the regular forms.

Indic & Indic-derived Scripts: Spaceless Scripts

มนุษย์ทั้งหลายเกิดมามีอิสระ ... *Thai*

လူတိုင်းသည် တူညီ လွတ်လပ်သော *Myanmar*

ມະນຸດເກີດມາມີເສຣີພາບ ... *Lao*

A number of scripts of Southeast Asia do not place spaces between individual words, although spaces may occur between phrases and sentances.

Indic & Indic-derived Scripts: Spaceless Scripts

- **Thai**
- **Lao**
- **Khmer *(Cambodian)***
- **Myanmar *(Burmese)***
- **Lanna *(added to Unicode in 2006)***
- **Tai Dam *(Viet Tai, Unicode proposal exists)***
- **Tai Le *(Dehong Dai, now in Unicode)***
- **New Tai Le *(now in Unicode)***

Spaceless scripts include Thai, Lao, Khmer, Myanmar, Lanna, Tai Dam, Tai Le, and New Tai Le.

Indic & Indic-derived Scripts: Spaceless Scripts

Allhumanbeingsarebornfreeandequal

English

Fussballweltmeisterschaftsqualifikationsspiel

German

People coming from other script traditions may wonder whether spaceless scripts are difficult to read.

Removing the spaces from an English sentance convincingly demonstrates that if you already know a language, it is not that difficult to read without spaces between words.

Long compound words in German represent a similar phenomenon in a western European language.

Just as line breaking and hyphenation are important concerns in western typography, word breaking and syllabification are equally important issues in typography for the spaceless scripts of Southeast Asia.

World Typography

- Text Layout Features of Scripts
  - *Indic & Indic-derived Scripts*
  - *Middle East & African Scripts*
    - Arabic Script
  - *Asian Scripts*
  - *Multiple World Scripts*

THE LINUX FOUNDATION

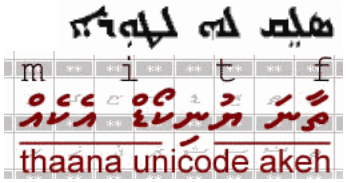
Let’s now look at some features of Middle Eastern scripts as well as one African script.

Middle East & African Scripts: Right-to-left Directionality

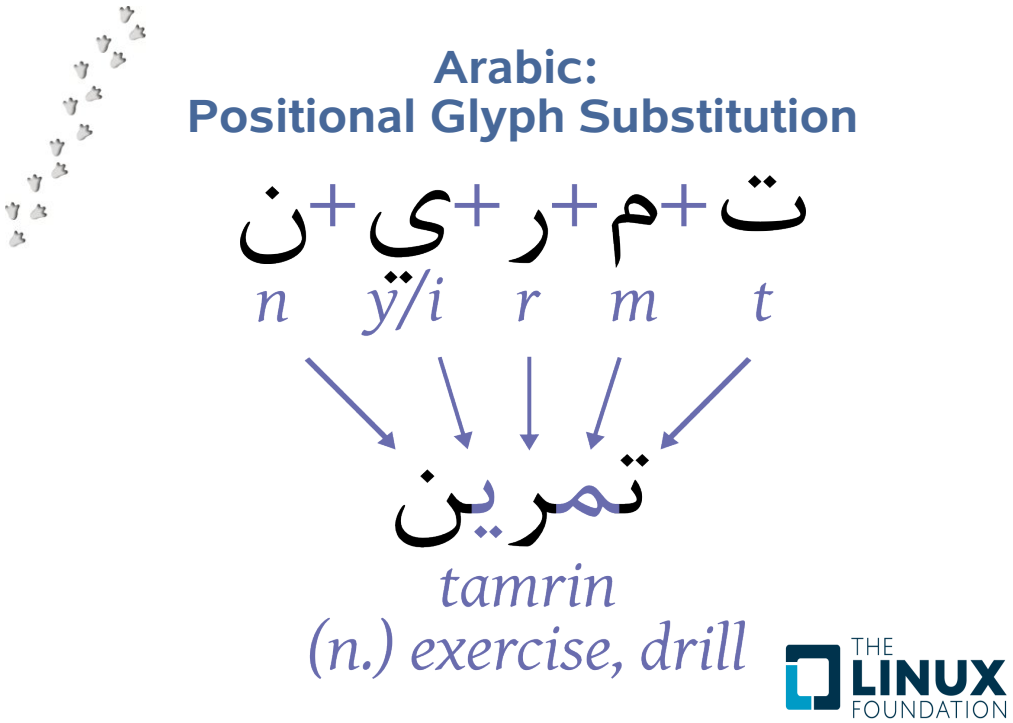
- Arabic
- Hebrew
- Syriac
- Thaana
- N'Ko

أهلاً وَسَهلاً

לֹא תַעֲשֶׂה־לְךָ פֶּסֶל וְכָל־תְּמוּנָה



A well-known characteristic of Middle Eastern scripts is that they are read from right to left (RTL). The African N'Ko script is also an RTL script.



Arabic, like Syriac, is an example of a cursive script where letters join to one another.

When letters are joined, the shapes of letters often change depending on where the letters appear in a word.

In the example shown here, you can see examples of *isolated*, *initial, medi*al, and *final* forms.

Note that the letter “r” (*reh*) in the middle of the word “tamrin” is a *final* form because this letter does not have a medial form. The *yeh* that follows is therefore in *initial* form.

Arabic: Mandatory Ligatures

ا + ل

alef *lam*

↓

لا

lam-alef ligature

THE LINUX FOUNDATION

The letters *lam* and *alef* form a mandatory ligature in Arabic.

Arabic:
Optional Stylistic Ligatures

Arabic OpenType font with set of extended ligatures

Arabic OpenType font without set of extended ligatures

تمرين ← تمرين

Even though *lam-alef* is the only mandatory ligature, it is quite common to see many other ligated forms in higher-quality Arabic typography.

These ligated forms represent attempts to preserve the highly-developed calligraphic forms of the script within the confines of technologies – first the typewriter and then the computer – that were not really designed with Arabic in mind.

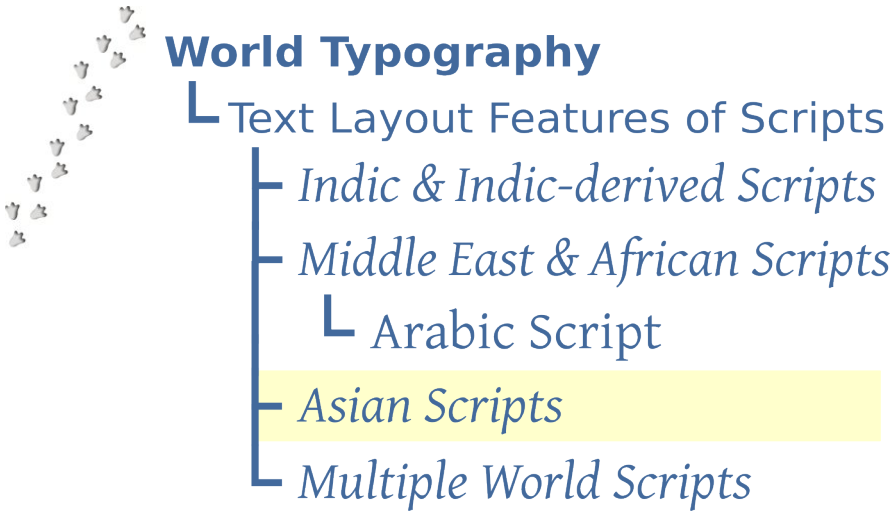
Arabic: Kashida

بســـــم الله الرحمن الرحيـــــم

Kashida is a method of justification in Arabic or Syriac in which *tatweel* characters are inserted at certain points in words in order to stretch them to fill out a line of text.

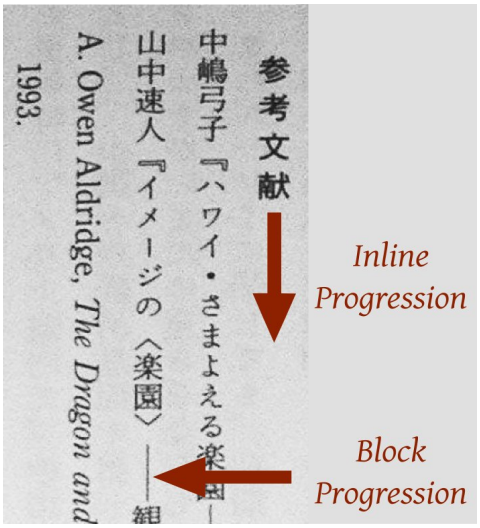
The contrast in the calligraphic and typical typeset appearance of the *bismi Allah* illustrated here effectively demonstrates the limitations of modern typesetting technologies for Arabic.

While the calligraphy (*top*) dances gracefully across the page, the typeset version at the bottom proceeds in a monotonous staccato.



Let’s look at some unique characteristics of Asian scripts.

Asian Scripts: Vertical Layout – Japanese & Chinese

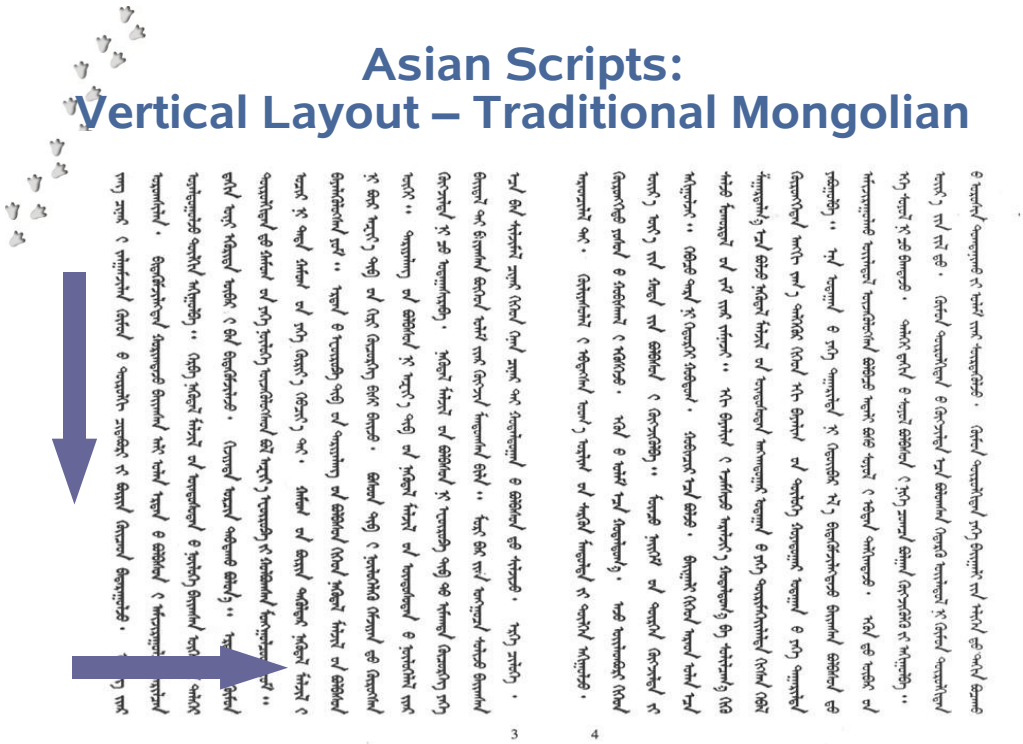


Japanese, Traditional Chinese, and Traditional Mongolian are quite commonly typeset vertically.

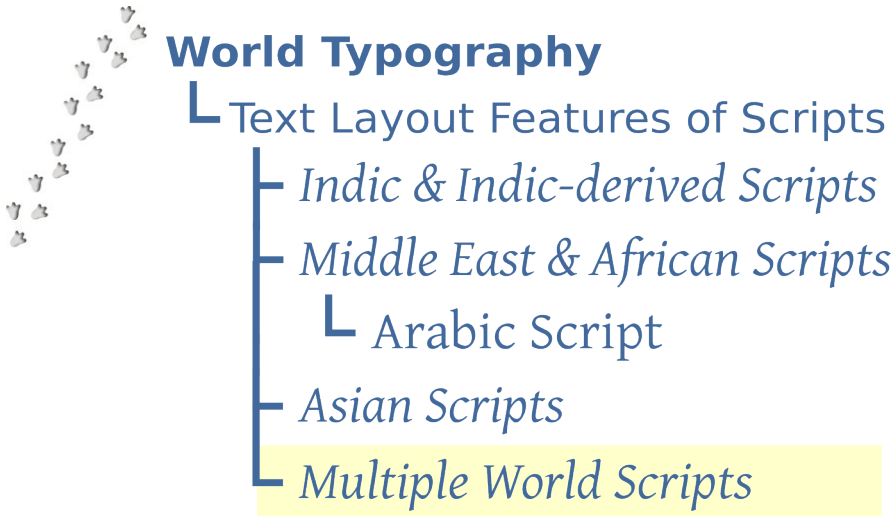
Japanese and Traditional Chinese are typeset from top to bottom with block progression running from right to left.

Horizontal layout of Chinese and Japanese is a modern invention.

Asian Scripts:
Vertical Layout – Traditional Mongolian



Block progression in Traditional Mongolian runs uniquely from left to right.



Finally, let's look at a few features of text layout that are common to numerous scripts, as well as phenomenon that occur when more than one script appears in a single "mixed" text – an increasingly common phenomenon in the modern world.

Multiple World Scripts: Diacritical Marks

- **Diacritical Marks**
- **Vowel Marks**
- **Cantillation Marks**
- **Tone Marks**
- **etc. ...**

Diacritical marks, vowel marks, cantillation marks, tone marks and various other symbols are commonly written above or below letters in numerous world scripts.

Multiple World Scripts: Baseline Adjustments

A. No baseline adjustment: inconsistent interline spacing

Beginning in the 1920s and 1930s, Marc Aurel Stein and others found thousands of scrolls in the Mogao Caves 莫高窟 near Dunhuang 敦煌. They included over 50 partial and complete Tao Te Ching 道德經 manuscripts. One written by the scribe So/Su Dan 素統 is dated 270 CE, and corresponds closely with the Heshang Gong 河上公 version. Another partial manuscript has the Xiang'er 想爾 commentary, which had previously been lost.

B. Baseline adjustment of Chinese: consistent interline spacing

Beginning in the 1920s and 1930s, Marc Aurel Stein and others found thousands of scrolls in the Mogao Caves 莫高窟 near Dunhuang 敦煌. They included over 50 partial and complete Tao Te Ching 道德經 manuscripts. One written by the scribe So/Su Dan 素統 is dated 270 CE, and corresponds closely with the Heshang Gong 河上公 version. Another partial manuscript has the Xiang'er 想爾 commentary, which had previously been lost.

THE LINUX FOUNDATION

In text containing a mixture of various scripts, it is often desirable to adjust the baseline positioning of the letters in one script relative to the other script. When adjustment is not performed, as shown on the left, sub-optimal typography results.

World Typography: Text Layout Requirements Summary

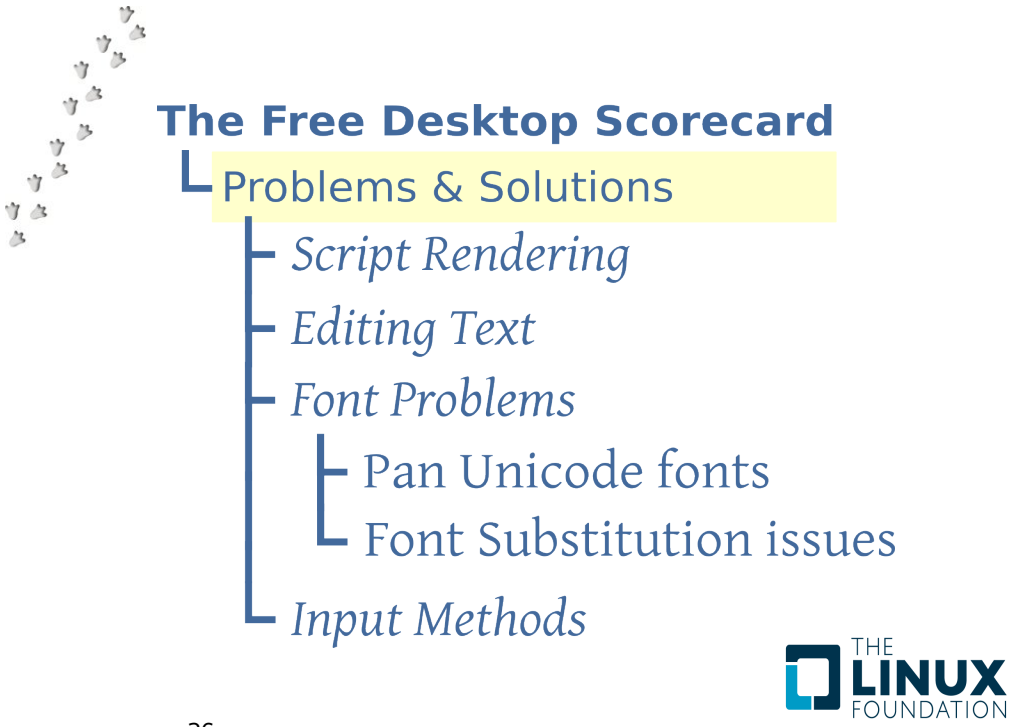
- **Glyph Substitution**
- **Glyph Positioning**
- **Baseline Adjustments**
- **Vertical Metrics**
- **Stylistic Alternates**
- **Optional Ligatures**
- **Swash Forms**

Scripts such as Arabic and Devanagari require mandatory glyph substitutions and glyph rearrangements. In mixed-language texts, baseline adjustments may be required. Vertical metrics are needed for Japanese, Traditional Chinese, and Traditional Mongolian text layout. Stylistic alternates, optional ligatures and swash forms are required for modern professional typography in the West as well as in the East.

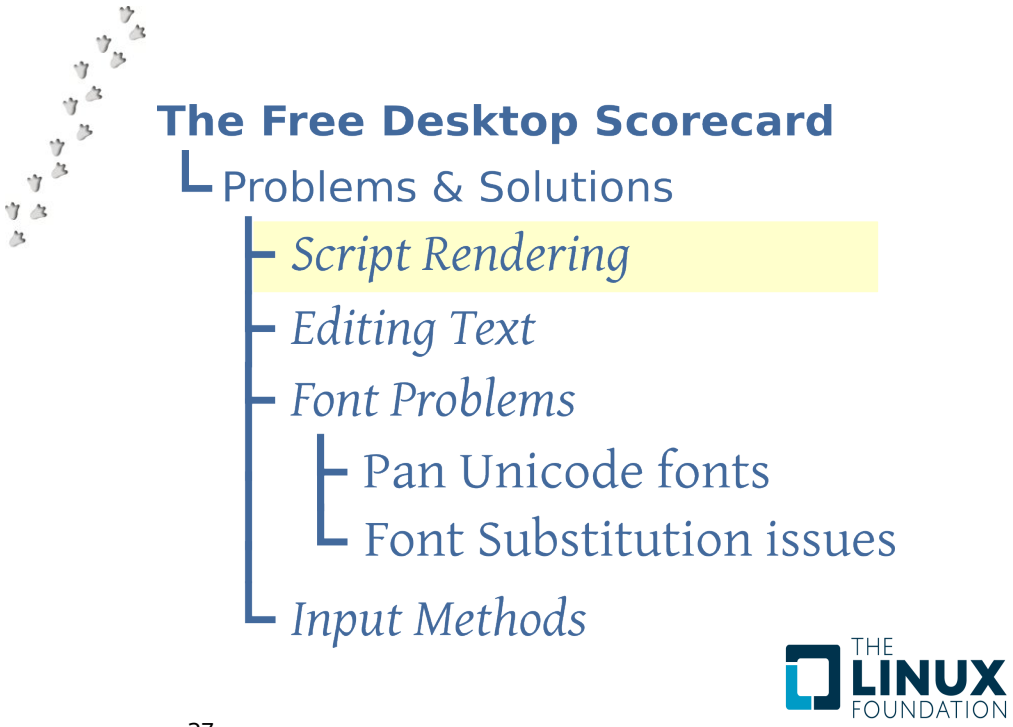
Free Desktop Scorecard: Problems & Solutions

Now that we have an overview of the features required for World Typography, let's examine how well the Free Desktop scores.

Let's look at what are the problems, evaluate those problems, and look at some solutions.



We can divide our investigation into script rendering, editing text, font problems, and input methods.



We'll start with script rendering issues and solutions.

Script Rendering Support : Problems

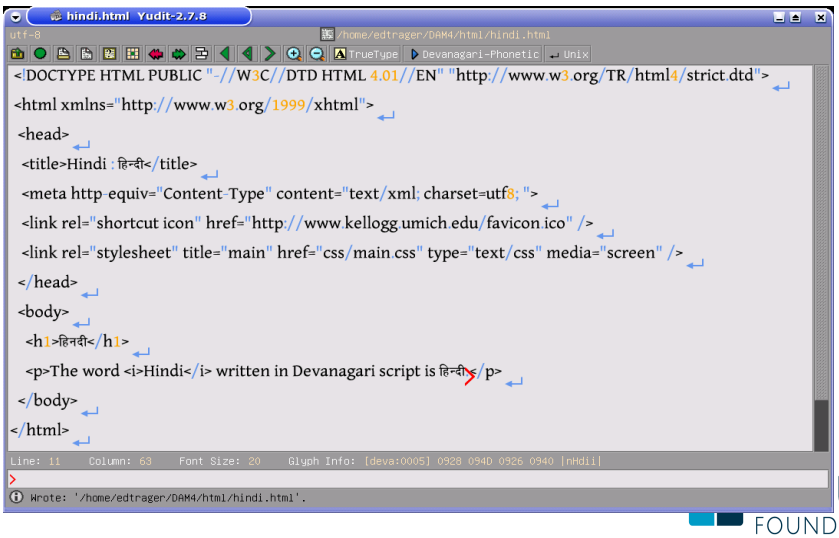
- **Inconsistent Support – different layout engines**
- **Buggy Support – esp. Indic**
- **Inefficient Support**
- **Not Supported At All – Myanmar, Mongolian ...**

Let's first look at how well world scripts are supported on the Free Desktop.

Problems include inconsistent support across different programs due to the use of different layout engines on the backend, buggy support especially for Indic scripts, inefficient support, and scripts that are not supported at all.

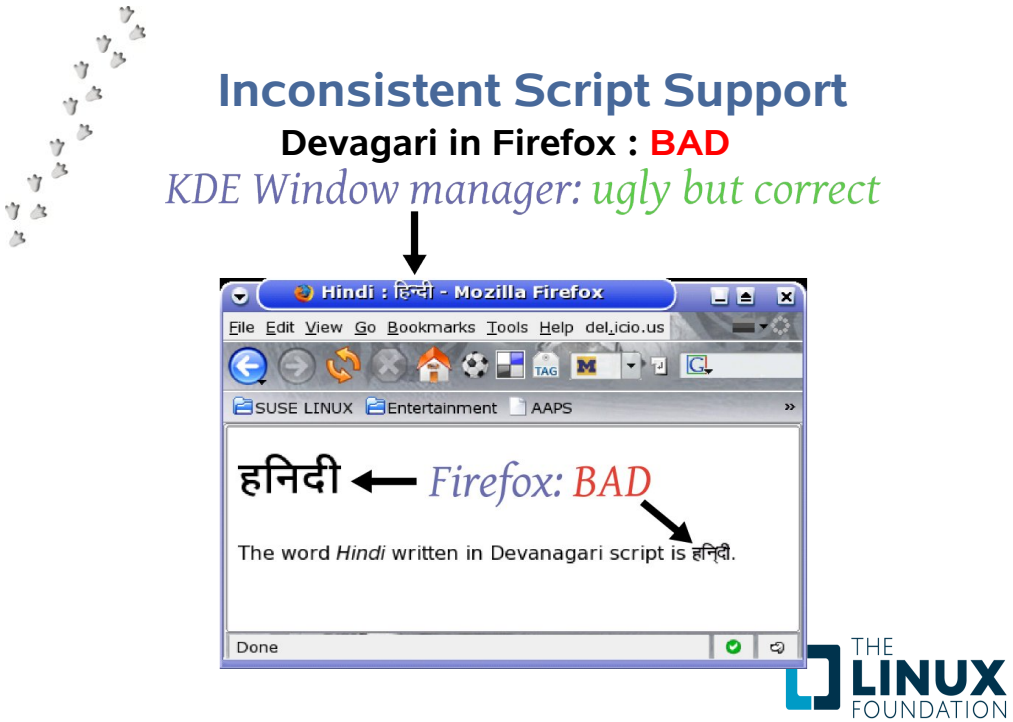
Inconsistent Script Support

Devagari in Yudit : Good



A huge problem on the Free Desktop is that support for various world scripts is inconsistent across different programs. This occurs because QT/KDE, GTK/Gnome, Open Office, and stock builds of Firefox all use different shaping engines for processing text. (QT uses Qt's engine, GTK/Gnome use Pango, Open Office uses IBM's ICU text shaper, and stock Firefox uses built-in "Pango Lite.").

Let's look at a simple example which proves the point graphically. Here is a simple XHTML page with the word "Hindi" written in Devanagari script. It has been typed into Yudit, one of the best editors for world script support, and saved in the default UTF-8 encoding.

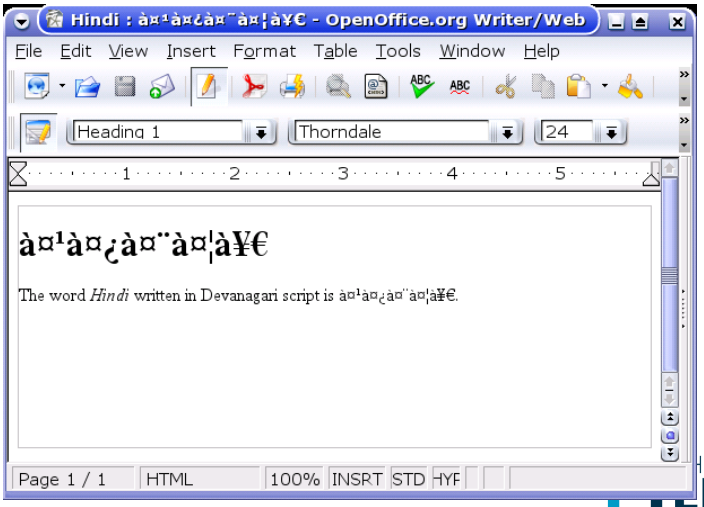


The title of the web page is rendered correctly by the KDE window manager; the result in the Gnome desktop would also be correct. (The result is technically correct but ugly – probably because a less-than-ideal font has been chosen for rendering the Devanagari. It is difficult to see in this screen shot, but there are alignment problems with the horizontal bar in the "nd" consonant conjunct glyph).

Unfortunately, the stock versions of Firefox downloaded from Mozilla.org do not use the local Pango libraries and therefore do not render Devanagari correctly. As shown in this screen shot from version 1.5+, neither glyph repositioning nor glyph substitutions are being performed. The problem persists in stock versions of Firefox 2.0+ as well.

Inconsistent Script Support

Devagari in OO Web : BAD – UTF-8 recognition

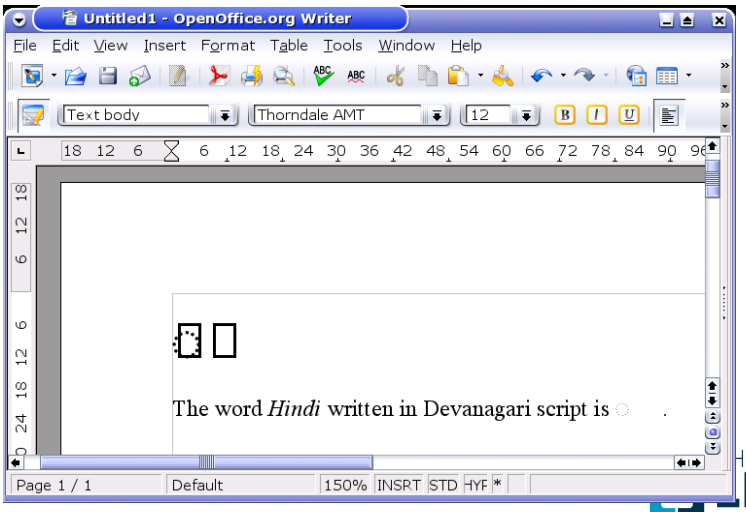


When I open up my *hindi.html* web page with Open Office, it appears that the editor assumes that web pages are encoded in ISO Latin-1 rather than in UTF-8 by default.

UTF-8 would be a better choice.

Inconsistent Script Support:

Devagari in OO Writer : BAD font substitution

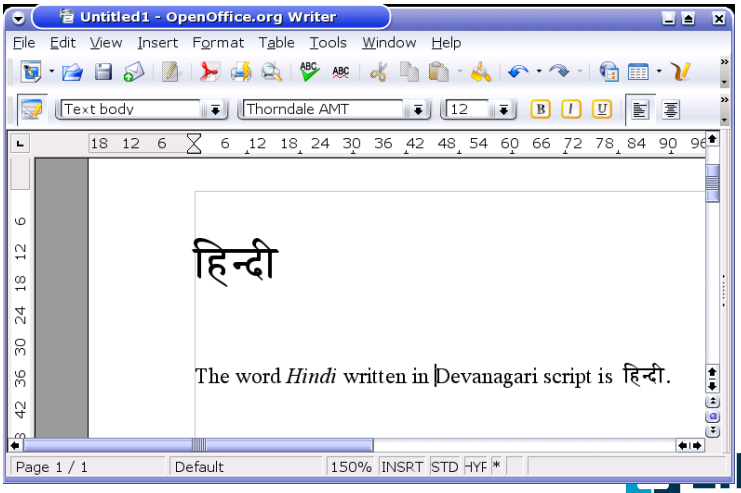


Since my test with the web editor failed, I decided to simply copy and paste the text into Open Office's writer program. Open Office 2.0 (Novell Edition on SuSE 10.0) fails to find an appropriate font for the Devanagari text.

Inconsistent Script Support

Devagari in OO Writer :

OK (After manually selecting Chandas font)



After I manually selected the *Chandas* font, I was finally able to see the text correctly in Open Office.

I of course have some experience dealing with non-Latin text on the Free Desktop. But the steps I took in my demonstration here are typical of what your average computer user might try to do. I think you will agree that the results are not exactly what the user would expect to encounter.

Buggy Script Support

- *On May 24, 2007, Behdad Esfahbod wrote on the HarfBuzz mailing list:*

"Note that the previous consensus that Qt's shapers are preferred was about the Indic shaper, not about them all in general. For Indic shaper, nobody knows which one (Pango, Qt, ICU) is the best. What we know is that maintainers of Pango and ICU are not happy about theirs. That's how it was decided to use Qt's. But at the end it really doesn't matter as the three are mostly the same, and each has its own share of bugs."

In addition to *inconsistent* script support, support for some scripts remains quite *buggy*.

HarfBuzz is an effort to create a unified text layout engine for the Free Desktop. We'll talk more about HarfBuzz in a minute.

On May 24, Behdad Esfahbod, one of the primary HarfBuzz architects noted on the mailing list that no one knows which of the existing Indic shapers (Pango's, Qt's, or ICU's) is best, that the Pango and ICU folks are in any case not happy about theirs, and that all of them have their own share of bugs.

Inefficient Script Support

- *On October 3, 2006, Trager to Esfahbod:*
- *Trager:* "In looking at the (Arabic shaper) code, I could not help noticing that first the UTF-8 string is converted to a UCS4 string ... but then in the subsequent for(;;) loop, the UTF-8 string is converted a second time, but this time character-by-character ..."
- *Esfahbod:* "Your observations are all correct."
- Ubuntu Bug #32561: "Pango-enabled Firefox is much slower"

Another problem is *inefficient* script support.

For example, in talking with Behdad Esfahbod about Pango's shaping engines, I noticed that the Arabic shaper was converting the UTF-8 string to UCS4 *twice*.

This provided some concrete evidence to me substantiating many people's complaint about Pango-enabled Firefox being slower.

Scripts Not Supported At All

- **Myanmar : 32.3 million speakers**
- **Classical Mongolian : 5 million speakers**
- **Lanna : 6 million speakers**
- ***etc. ...***

Compare to :

- ***Modern Greek : 12.2 million speakers***
- ***Georgian: 4.1 million speakers***

Finally, there is the problem that *some scripts are still not supported at all.*

Shaping for Myanmar, for example, is not yet included in Pango 1.17.2.

Classical Mongolian is not yet available in any of the Free Desktop text layout engines (Pango, QT, ICU).

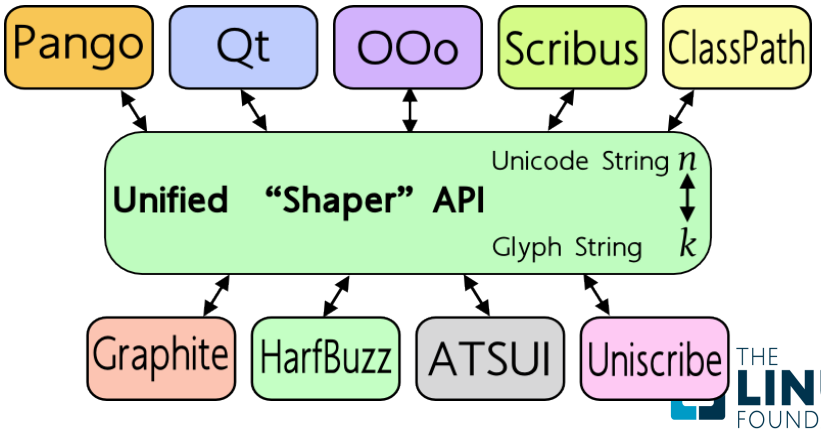
Lanna, similar to the old Mon script and approved for inclusion in Unicode just last year, is also not yet supported. Lanna's encoding model is similar to that of Khmer. To my knowledge, no roadmap or project yet exists to provide support for Lanna or for other scripts which will be entering Unicode in the near future.

Since it is not possible to determine the number of readers of these scripts, I have provided reliable estimates of the number of speakers instead. For comparison, the number of speakers of Modern Greek and Georgian are provided.

As you can see, there are more than twice as many speakers (and presumably therefore also readers) of Myanmar as there are speakers of modern Greek, but Myanmar remains unsupported.

Script Rendering Support : Solution

- Text Layout Summit @ Boston Gnome Live! 2006
- Agreement on unified “shaper” API
- “Low hanging fruit” is HarfBuzz OpenType Library



The solution to script rendering problems on the Free Desktop is to create a single, unified text layout library which incorporates the best facets and ideas of existing FLOSS layout engines (Pango, Qt, ICU, Graphite) and which may eventually also provide seamless integration with proprietary engines such as ATSUI and Uniscribe.

Agreement in principle on such a unified shaper API was achieved by participants at the first Text Layout Summit at the Gnome Live! meeting in Boston in October, 2006. A chalkboard version of the diagram you see here was sketched out at that meeting.

Behdad Esfahbod’s HarfBuzz OpenType library was seen as the “low hanging fruit” where work on creating a unified text layout engine for the Free Desktop could quickly become a concrete reality.

Script Rendering : Word breaking & Syllabification of Spaceless Scripts

- **Problems:**
  - Software only available for Thai (libThai)
  - Little or no support for Khmer, Myanmar, Lao and others
- **Solutions:**
  - Create an object-oriented framework for rule- & dictionary-based word-boundary analysis
  - Plug into unified text rendering pipeline

มนุษย์ทั้งหลายเกิดมามีอิสระ ... *Thai*

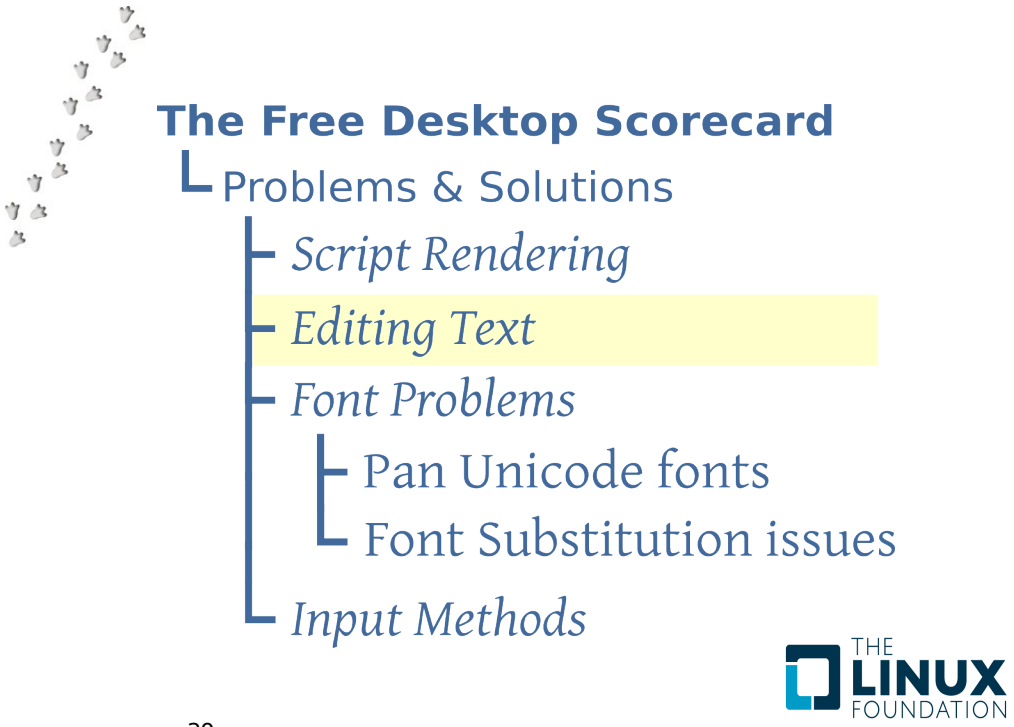
လူတိုင်းသည် တူညီ လွတ်လပ်သော *Myanmar*

ມະນຸດເກີດມາມີເສຣີພາບ ... *Lao*

Line breaking is an issue in both Western and Eastern typography. For spaceless scripts, a word breaking library is currently only available for Thai.

While developers have researched and written software to perform word breaking and syllabification tasks for other scripts like Khmer, Lao, and Myanmar, to my knowledge such software has not yet become incorporated into standard libraries available on the Free Desktop.

A solution would be to create an object-oriented framework for both rule- and dictionary-based word-boundary analysis which could then be plugged into a unified text-rendering pipeline.



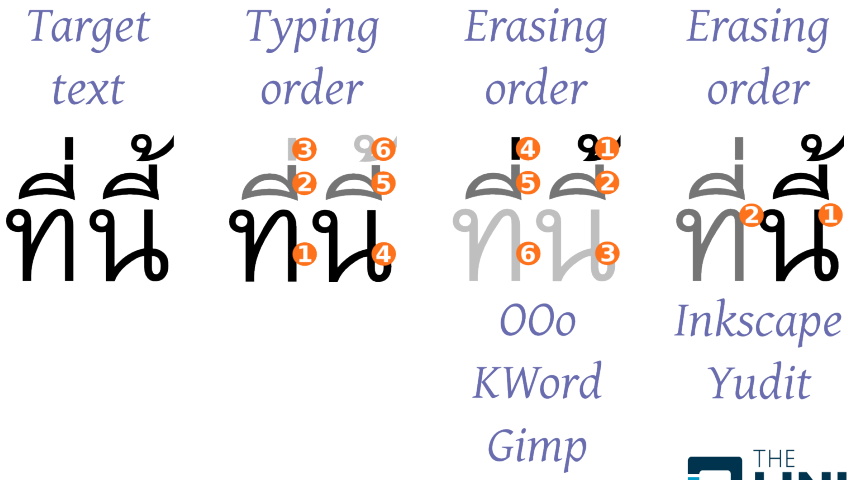
Editing text – especially complex script text like Devanagari or Arabic – can also be an issue on the Free Desktop.

Editing Text : Problems

- **Backspace or delete gobbles entire cluster**
- **Cursor positioning makes no sense**

One often-encountered problem is that backspacing or deletion erases an entire cluster rather than just the last diacritical or vowel mark. Cursor positioning can also be a problem.

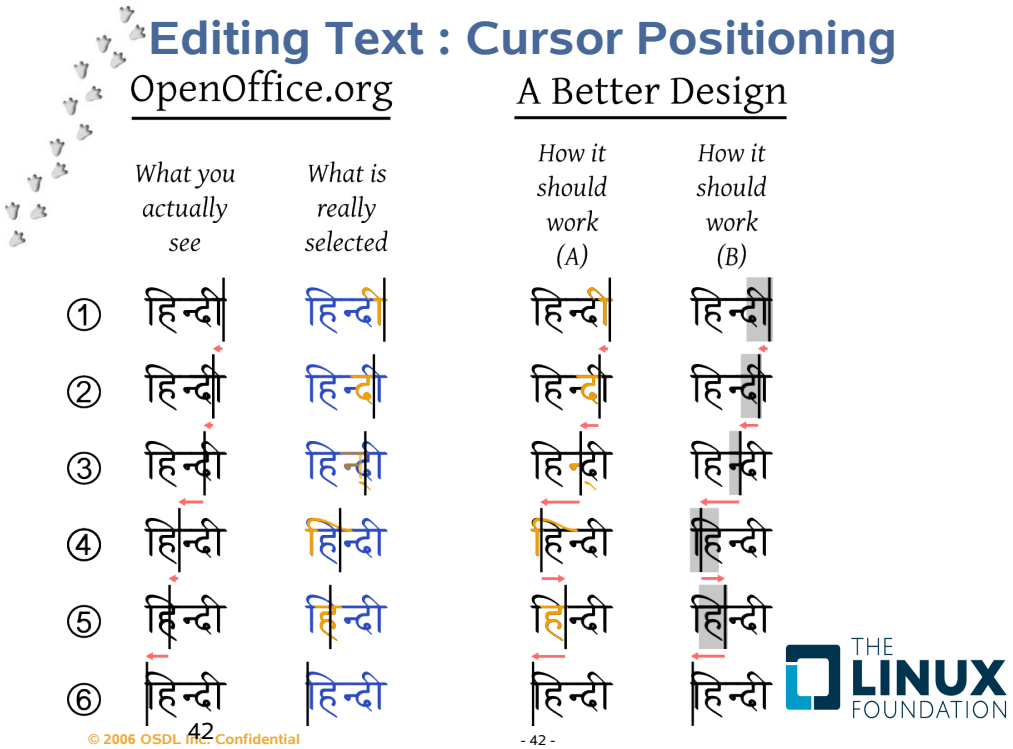
Editing Text : Delete Gobbles Cluster



A concrete example in Thai illustrates the problem.

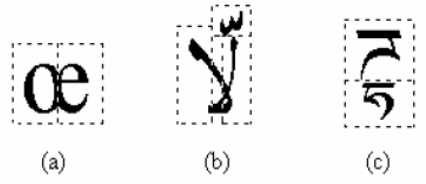
To obtain the text on the left requires typing six characters, as shown in the second frame. In programs like Open Office, KWord, and the Gimp, the desired behaviour occurs when erasing : the diacritics (tone and vowel marks) are erased one glyph at a time in the reverse order in which they were typed above the base consonant.

However, in some other programs – like Inkscape – the entire consonant-plus-diacritics cluster is erased with a single keystroke. This behaviour is usually quite annoying, since it is not uncommon to simply type the wrong tone mark or vowel mark over an otherwise correct base consonant.



Cursor positioning is another problem for complex Indic scripts especially. The actual behaviour, as shown for OpenOffice on the left, is completely non-intuitive. Two better models are shown on the right.

Editing Text : Solutions



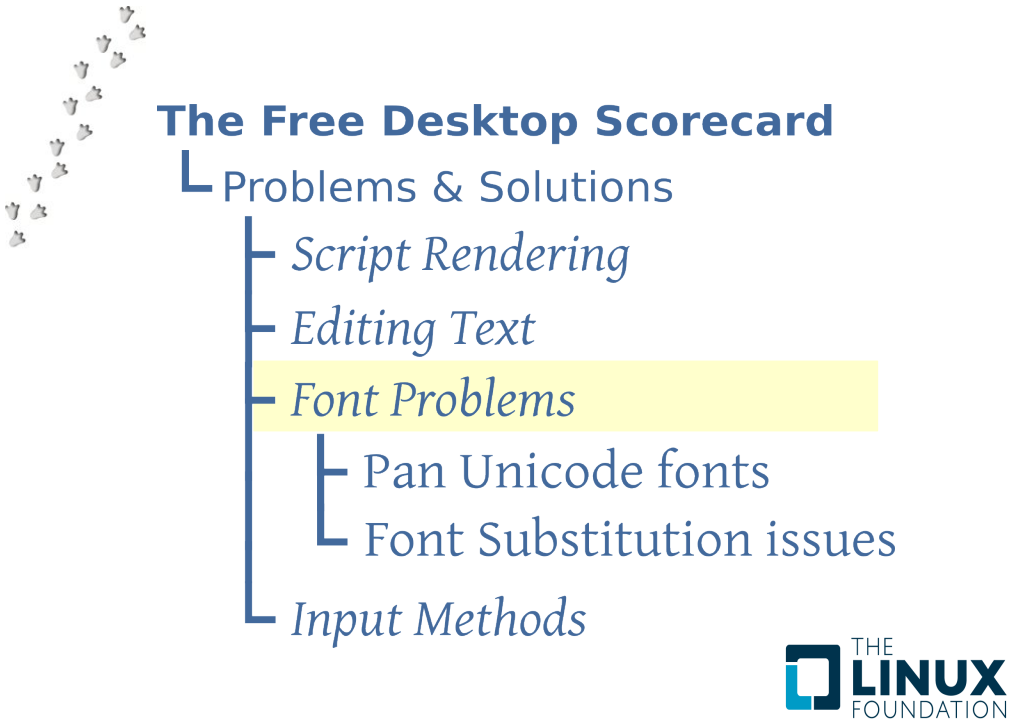
- **Ligatures are not monolithic**
- **Underlying characters are accessible**
- **Access rectangular areas corresponding to glyph components**
- **Highlighting & mouse selection handled by text layout engine (Graphite, HarfBuzz, etc)**
- **No complex programming required**

THE LINUX FOUNDATION

Solving these text editing problems requires a system where ligatures are not monolithic, where the underlying characters are accessible, and where access to the rectangular areas corresponding to the glyph components are also available.

SIL's **Graphite** engine has these capabilities. The **HarfBuzz** library will likely have a similar set of capabilities.

It would very beneficial if capabilities such as these were accessible from a well-designed and simple-to-use unified API for the Free Desktop.



A number of font problems also exist on the Free Desktop.

Font Problems : Pan-Unicode Fonts

- **We don't need Pan-Unicode fonts because:**
  - An increasing number of high-quality fonts for specific scripts are being produced by dedicated groups
  - Flexibility of fontconfig allows construction of virtual fonts that cover Unicode code space better
- **But groups persist in adding more Unicode blocks to fonts:**
  - Many users were not happy when Arabic first added to DejaVu
- **Asian CJK fonts also often have inferior Latin/Greek/Cyrillic (LGC) glyphs**

Several factors have emerged in recent years which, in my mind, have greatly reduced the perceived necessity and utility of the Pan-Unicode fonts.

First, an increasing number of free and liberally-licensed Unicode fonts of very high quality for specific scripts and specific uses (such as scholarly) have been released by dedicated groups and individuals. The existance of George Williams' Open Source FontForge outline font editor has certainly played an important role in the democratization of typography. One need only take a cursory look at some of the excellent work being done by a diverse spectrum of organizations --like SIL International, The Tibetan & Himalayan Digital Library, or the Khmer Software Initiative-- and individual people --like Christopher Harvey's work on fonts for native American languages at LanguageGeek.com, Paul Morrow's work on Babayin at Sarisari Filipino History, or Firefly and Qianqian Fang's contributions to Open Source Chinese fonts-- to realize how true this is.

Secondly, the flexibility of Keith Packard's Fontconfig library allows the construction via simple XML-based configuration files of virtual font sets (such as "sans" and "serif") which can do a better job than any one Pan-Unicode font can at covering the Unicode code space with high-quality glyphs coming out of projects such as those highlighted in the previous paragraph.

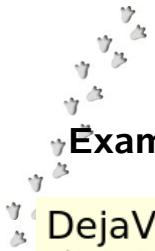
Some projects seem to still be searching for their identity and continue to add Unicode blocks with unexpected results. When the DejaVu project first added Arabic glyphs, many users were unhappy. In the end, the DejaVu team decided to release a separate “LGC” (Latin-Greek-Cyrillic) font set that no longer contained the disputed Arabic glyphs.

Another kind of “pan-Unicode” font problem occurs with CJK fonts that, by and large, contain inferior glyphs for Latin, Greek, and Cyrillic.

Font Solutions

- **Use specific fonts for specific orthographies**
- **Extend Fontconfig syntax to be able to specify fonts by script and orthography directly**

The solution should be to use specific fonts for specific orthographies and extend fontconfig's syntax to be able to specify fonts by script and orthography directly. The next four slides cover two examples.



Font Solutions

Example: Vietnamese expect more vertical spacing:

DejaVu Sans

Đời cha ăn mặn,
đời con khát nước. 7pt.
Đời cha ăn mặn,
đời con khát nước. 8 pt.
Đời cha ăn mặn,
đời con khát nước. 10 pt.
Đời cha ăn mặn,
đời con khát nước. 12 pt.

VU Pho Tho

Đời cha ăn mặn,
đời con khát nước. 7 pt.
Đời cha ăn mặn,
đời con khát nước. 8pt.
Đời cha ăn mặn,
đời con khát nước. 10 pt.
Đời cha ăn mặn,
đời con khát nước. 12 pt.

THE LINUX FOUNDATION

Here is an example: Vietnamese users expect more vertical spacing. Fonts designed for the Vietnamese market provide that spacing. “All purpose” fonts like DejaVu do not.

Vietnamese readers are also very particular about the appearance of the diacritical marks which are critical components of their written language. A font like VU Pho Tho may therefore be preferred over DejaVu Sans for multiple reasons.

Font Solutions

... so extend fontconfig syntax to specify fonts by script and orthography directly:

```
<script>
  <name>latn</name>
  <orthography>
    <name>vn</name>
    <alias>
      <family>VU Pho Tho</family>
      <family>DejaVu Sans</family>
      <default><family>sans</family></default>
    </alias>
  </orthography>
</script>
```


The extended XML syntax shown here would make it trivially easy to prefer VU Pho Tho for Vietnamese orthography.

Using the syntax shown here, one could specify a set of default fonts for the *latn* script which would be used when no additional locale or language information were available. One could also have as many nested *orthography* clauses as desired for specify specific font choices for specific orthographies.

Font Solutions

Japanese & Chinese prefer different fonts:

直骨人 AR PL ShanHeiSun Uni 中
直骨人 AR PL New Sung 中
直骨人 cwTeXMing 中
直骨人 Sazanami Mincho 日
直骨人 IPAPGothic 日
直骨人 Konatu 日

A second example: Chinese (中) and Japanese (日) users have non-overlapping font preferences. Current fontconfig XML syntax rules do not provide an easy solution.

Font Solutions

Extended XML syntax would solve the problem:

```
<script>
  <name>hani</name>
  <orthography>
    <name>zh</name>
    <alias>
      <family>AR PL ShanHeiSun Uni</family>
      <default><family>serif</family></default>
    </alias>
  </orthography>
  <orthography>
    <name>ja</name>
    <alias>
      <family>Sazanami Mincho</family>
      <default><family>serif</family></default>
    </alias>
  </orthography>
</script>
```


The extended fontconfig XML syntax suggested here would make it trivially easy to specify different fonts for Chinese and Japanese users.

Configuring Fonts for Screen & Print

- **Problems:**
  - Fontconfig only defines serif, sans, monospace categories
  - Appropriate only for Western typography
- **What additional categories would be most useful?**
- **Solutions:**
  - ***Screen*** category
  - ***Print*** category

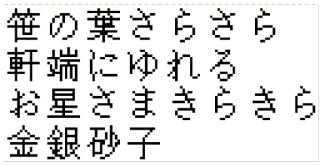
Fontconfig by default only defines serif, sans, and monospace categories. In many ways, these categories only make sense in the realm of Western typography.

(In a World Typography context, it might perhaps be better to say *modulated* instead of *serif* and *unmodulated* instead of *sans*.)

For many scripts, having default *screen* and *print* font categories would be very useful.

Configuring Fonts for Screen & Print

Screen - small to medium pixel sizes:
GNU Unifont or Wen Quan Yi bitmap font



Print: IPA Mincho

笹の葉さらさら
軒端にゆれる
お星さまきらきら
金銀砂子

Screen - larger pixel sizes: IPA Gothic

笹の葉さらさら
軒端にゆれる
お星さまきらきら
金銀砂子

Users of Japanese and Chinese in particular would benefit.

Because computer screens have fairly low resolution and Japanese and Chinese contain many strokes, users often prefer a bitmap font for reading kanji and kana at small to medium sizes on screen.

At somewhat larger pixel sizes, an unmodulated Japanese “Gothic” or Chinese “Hei” style font is appropriate.

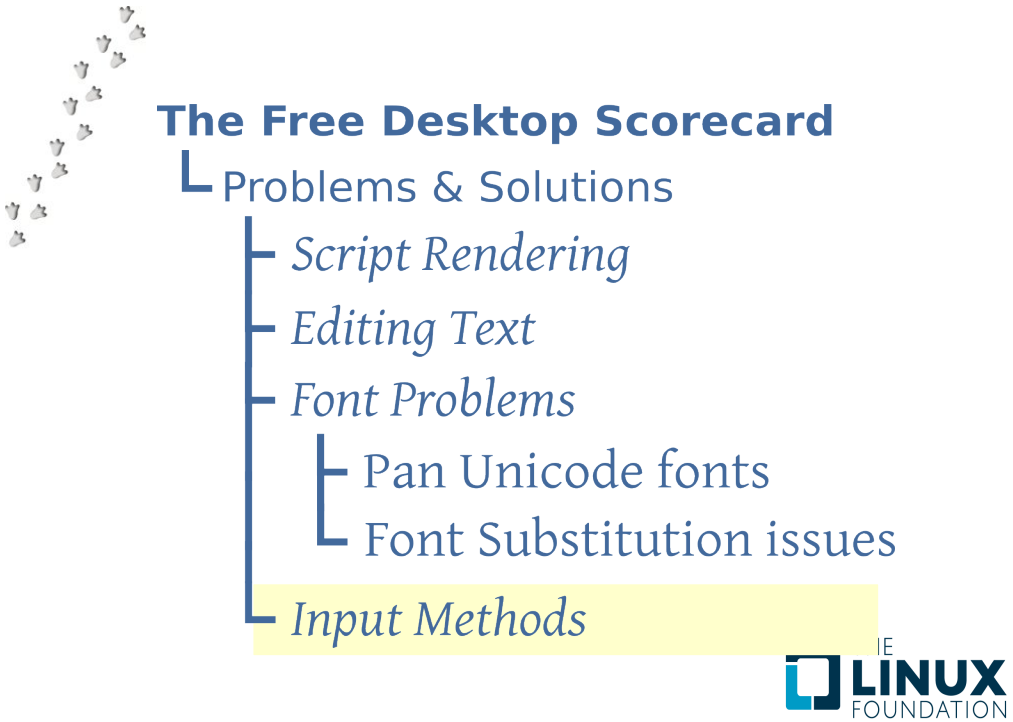
In contrast, Japanese is almost always printed using a “Mincho” style font. For Chinese, the equivalent would be a “Song” style font. This is similar to the way Westerners generally prefer a “Roman” font for printed documents but may use a *sans-serif* font for greater clarity on screen.

Font Solutions : Fostering an Open Typography Community

- **Problems:**
  - "Free font" typographers historically worked independently, little cooperation
  - Often not aware of licensing issues
  - Often used unclear, ad-hoc, or incompatible licenses
- **Solution:**
  - **SIL's Open Font License** encourages an Open typographic community to spring up and grow.

Historically "free font" typographers worked independently and there was little formal communication or cooperation. Font producers were often not aware of licensing issues and thus released fonts with unclear or ad-hoc licenses which failed to state whether derivative works could be produced.

SIL has now released the Open Font License which specifically addresses these concerns and encourages a truly Open typographic community to spring up and grow.



A number of font problems also exist on the Free Desktop.

Input Methods

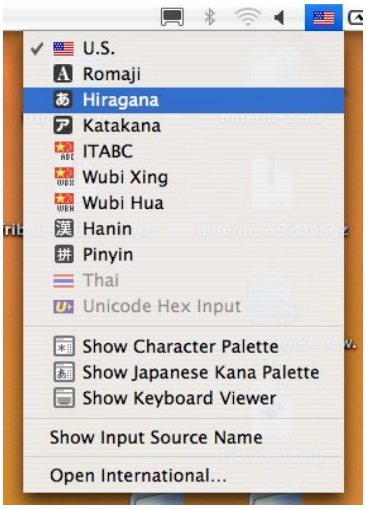
- **Problem:**
  - Keyboard Layouts & Input Methods not unified

KDE Keyboard tray access *SCIM IME tray access*

On the Free Desktop, keyboard layouts and input methods are often not unified. In addition, a number of different groups are currently working on input method issues, including m17n, SCIM, and KMFL.

Input Methods

- **Solutions:**
  - Streamline & unify access to keyboard layouts and IMEs on the desktop



Better integration of effort between these groups and the main desktop projects (KDE, Gnome) could result in streamlined, unified access to keyboard layouts and input methods on the Free Desktop.

The example here is from the Mac OS X desktop.

Wrap Up

Opportunities for the FLOSS community in the development pipeline

- **2nd Annual Text Layout Summit @ aKademy in Glasgow July 4-6, 2007**
- **Make unified text rendering pipeline a reality**
- **Increasing collaboration with non-traditional partners is already proving very fruitful**
- **Have a roadmap to implement new scripts as they are approved in Unicode**
- **Take a more pro-active role in pushing the Unicode Standard & World Typography on the Free Desktop to the next milestone.**

THE LINUX FOUNDATION

In summary, I've tried to cover most of outstanding issues in this presentation. Numerous opportunities exist for the FLOSS development community to participate in making the Free Desktop a better place for World Typography.

The upcoming Text Layout Summit will be held in conjunction with aKademy July 4-6 in Glasgow, UK. We anticipate continued progress on HarfBuzz to make a unfied text rendering pipeline a reality.

Increasing collaboration between FLOSS developers and non-traditional partners like SIL and the Open Typography community is already proving very fruitful and we can expect to see more exciting progress in the future.

I personally would like to see the FLOSS community develop a roadmap, in conjunction with partners such as SIL and the Script Encoding Initiative at Berkeley, to implement new scripts on the Free Desktop as soon as they are approved in Unicode.

There are many areas where the FLOSS community can take a more pro-active role in pushing the Unicode Standard and World Typography forward on the Free Desktop.

Resources

- **An Introduction to Indic Scripts by Richard Ishida (2003)** http://people.w3.org/rishida/scripts/indic-overview/
- **Free Desktop Text Layout Working Group** http://freedesktop.org/wiki/TextLayout
- **International Text Layout & Typography: The Big And Future Picture by Ed Trager** http://www.unifont.org/textlayout/TheBigPicture.pdf
- **HarfBuzz Mailing List** http://lists.freedesktop.org/mailman/listinfo/harfbuzz
- **Open Font License** http://scripts.sil.org/OFL
- **Unifont.org** http://www.unifont.org

Resources