回顧 GNU/Linux 中文資訊化 進展與未來機會

Jim Huang (黃敬群)

Developer, 0xlab

jserv@0xlab.org

Oct 12, 2010

Rights to Copy

http://0xlab.org/

contact@0xlab.org

creative commons
COMMONS DEED

Latest update:Oct 13, 2010

從一個笑話談起

三男子去女方提親，
說各自的經濟情況

A 說：「我有 2000 萬」

B 說：「我有一棟豪宅，
價值 5000 萬」

女方家長很滿意，就問
C：「你家有什麼？」

C 答：「我什麼都沒有，只有一個孩子，現在孩子在你女兒的肚子裡」

A、 B 無語，走了

啟示：

核心競爭力不是錢和房子，要有自己的人，在「關鍵位置」上

反過來想，立足於今日的資訊世界，難免會有力不從心的感嘆，但我們何嘗不在「關鍵位置」上呢？

中文資訊處理！

Agenda

(1) CLE 歷史背景與階段性目標

(2) GNU/Linux 中文資訊化焦點

資訊交換規範、輸出架構、輸入法、印列排版等

(3) 區域性自由軟體 / 國際化發展模式

(4) 未來的機會

平台整合與關鍵元件的分工、考量到移動裝置與雲端

CLE 歷史背景與階段性目標

CLE 歷史背景

- 在歷史故事之前，談 CLE 立意
  - Chinese GNU/Linux Extensions
  - 中文 Linux 延伸套件
- 一堆中文相關軟體的集合
- 「將來有一天，大家只要拿起各家原版 Linux 發行套件裝上，便有了基本的中文環境。到那時候， CLE 也許就可以功成身退了」(CLE v0.9p1 文件)
- 相似特性
  - 台灣地區的「國民大會」
  - 巴枯寧「國家制度和無政府狀態」

CLE 歷史背景

- 資料來源
  – http://cle.linux.org.tw/wp/about/
- 最初小虫只是把一些中文相關的軟體加上 patch，製作成 RPM 檔，目的只是為了方便安裝工作站
- 包出來的中文 RPM 檔越來越多，也發覺利用 RPM 系統把這些中文相關軟體作一個整合並且包裝成一組安裝套件，可以節省許多一一設定的時間

CLE 開發進程

- 1998 年 6 月 20 日發表 CLE **v0.3**
- CLE **v0.4** → **v0.5** → **v0.6**
  - 支援 TrueType 字體的 CXWin，中文輸入程式 xcin 2.3，終端機 crxvt，可處理中文排版的 ChiTeX，列印中文的 bg5ps
  - CLDP 計劃的中文文件、常見問題解答

當時的 CXWin 主要對 XFree86 3.x 作一系列 hack，使其能在 X font server 與 Xlib 可認得 Big5 字元 Xcin-2.3 多少也有如此的意味，雖可運作，但不穩定

- CLE 從小虫個人收集，慢慢成為真正開放原始碼的計劃，受到使用者的熱愛，成為台灣地區最多人使用的 Linux 安裝套件， CLE 的社群也逐漸形成

CLE 開發進程

- CLE **v0.7** (1999 年一月) 是成立通信論壇後眾人通力合作的第一個作品
  - 嘗試加入最基本的 I18N 支援，可關閉 CXWin
  - 加入顯示 / 輸入大陸 GB 碼的能力，統合華文地區 Linux 系統

時值 GNU libc5 → libc6(改稱 glibc2.x) 的變遷， CLE 之上一系列的軟體嘗試支援 locale extension

- CLE **v0.8** (1999 年六月):
  - 以 RedHat 6.0 為基礎，揚棄 CXWin ，改用外掛式 xa+cv
  - 以 glibc 2.1 為基礎的函式庫也加入中文區域化的支援，在少數支援 I18N 的程式中，可不依賴 xa+cv ，

XA = Xcin Anywhere， 是透過 LD_PRELOAD 機制去攔截 X 應用程式的輸出

CLE 開發進程

- CLE **v0.9** (2000 年三月)
  - 具多重象徵意義
  - 第一次以完整的 I18N 架構為主軸進行中文化，銜接國際化新技術
  - xcin 2.5： 使應用程式具有顯示 / 輸入中文能力
  - 「 CLE v0.9 不僅僅是一套『中文版 Linux 』，更是一套具有支援多國語言能力的 Linux 套件。除了內建支援的繁體及簡體中文外，只要再加上適當的區域化資料庫、字型、輸入法以及訊息翻譯， CLE v0.9 更可以輕易地變成日文版、韓文版等等」

技術背景可參考謝東翰 (居士) 的著作〈 Linux 的中文化問題簡介〉

也就是屏棄 xa+cv 的 hack， 全面採用 XIM 架構

CLE 開發進程

- CLE 現已功成身退
- (Nov 19, 2005) 居士：〈本人卸下 XCIN 計畫 coordinate 工作〉

  「過去 XCIN 計畫已在適當的時機填補了適當的空缺，算是完成了階段性的任務。今天，我們已有各式各樣五花八門的中文輸入軟體，諸如 gcin、 openvanilla、 iiimf、 scim、 libchewing ... 它們的設計無一不比 xcin 先進好用，大家也不需像從前那樣這麼依賴 xcin 了，這真的是一件好事，為此我感到非常高興。因此， XCIN 計畫是否應再維護下去？似乎也沒那麼必要了」

可是，當今中文資訊建設真的夠好了嗎？

考到及格就好了嗎？

- 胡適的觀點：
  - 「必須先用白話文字來代替文言的文字，然後把白話的文字變成拼音的文字」
  - 中國人「一分像人，九分像鬼的不長進民族」
  - 「象形文字的殘根餘孽能爬出中世紀的茅坑，多少算是救了漢字」
- 看看 Unicode IVD (Ideographic Variation Database) – Japan1-6
  - http://unicode.org/ivd/
- 面臨新的典範移轉 (paradigm shift)

GNU/Linux 中文資訊化焦點

另一個笑話

英文辭彙中

- **Lingual** 是口語或描述語言的用詞
- 會 (使用) 兩種語言稱為 **bi-lingual**
- 會 (使用) 三種語言稱為 **tri-lingual**

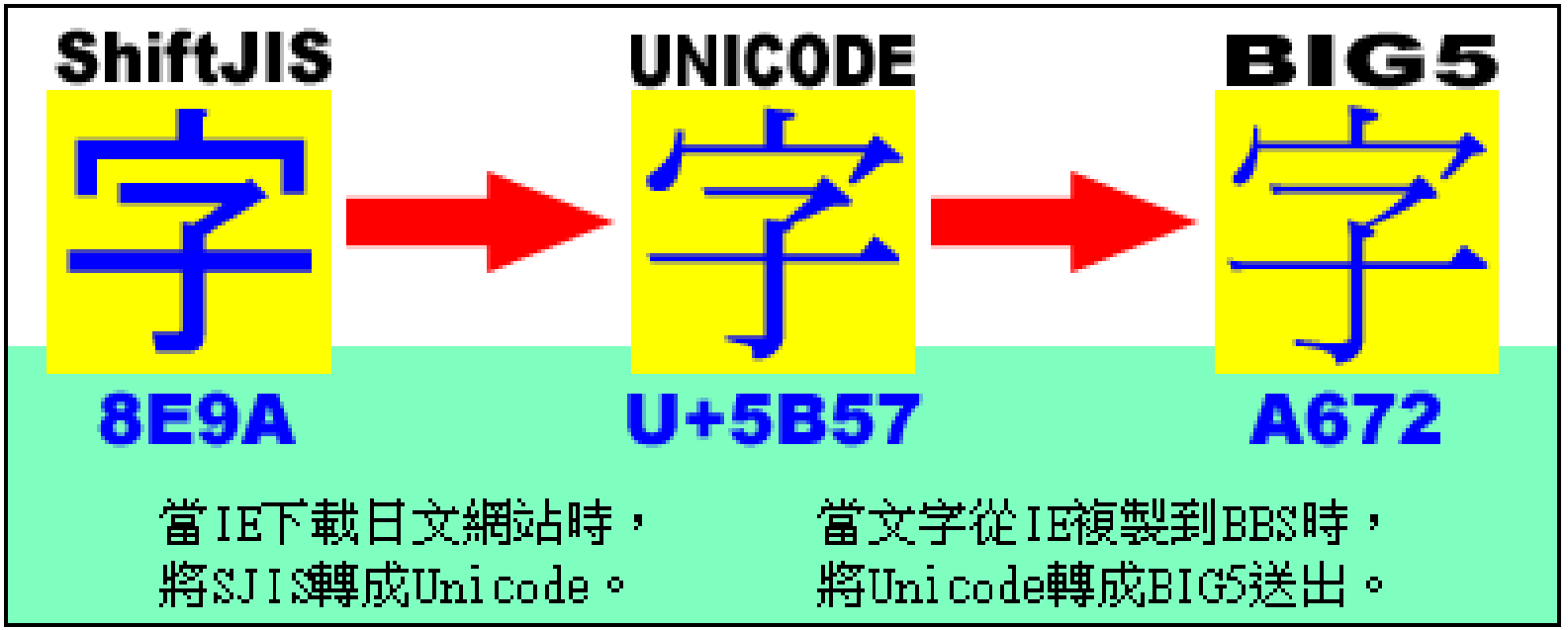
問：只會一種語言，該如何稱呼？

另一個笑話

- 英文辭彙中
  - 只會一種語言，該如何稱呼？
  - **American**
- 很諷刺，但 ...
  - 反映數位資訊化的過程，衍生的多國語文問題
  - 除 american 外，事實上許多數位資訊建設的使用者至少是 bilingual
  - 中文的資訊化過程，已面臨諸多議題
    - 廣義上，仍有如藏文資訊處理的議題

還記得那個萬碼奔騰的年代嗎？

- 其實一直存在
- 國際標準
  - ISO C90: Locales and Internationlization

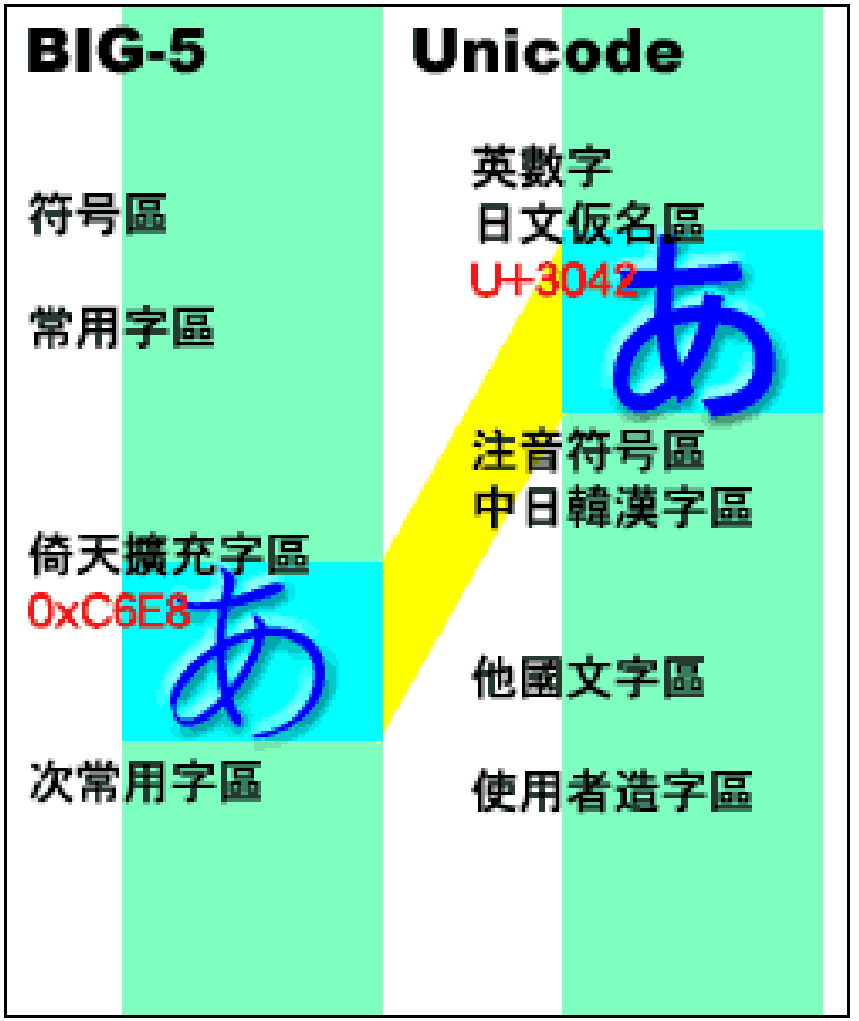


還記得那個萬碼奔騰的年代嗎？

- 無止息的「手術」

（前）

（後）



今日，「一分像人，九分像鬼的不長進民族」至少還得面對以下課題

- 「**國際化**」任務 (Internationalization, **i18n**)
  - 建立高度彈性的系統架構，可透過修改或調整，輕易的符合特定語言需求，更可以彼此交流
  - 語言的數位資訊化 / 書寫系統 (DBCS, bi-directional, …) / 匯率 / 曆法系統 / 時區或時間量測單位 / 度量衡 / 地理區域系統表示 / 郵遞區號或郵件系統
- 「**區域化**」任務 (Localization, **L10n**)
  - Locale = 語言 + 文化 (無法一言以蔽之)
  - 光英文就很多種 (US English / UK English / Chinese English / “Singlish”)
  - 國內有多種語言，瑞士官方語言：德文、法文、義大利文
  - 中文至少分繁 (正) 體與簡體，還不含地區差異

GNU/Linux 的實做

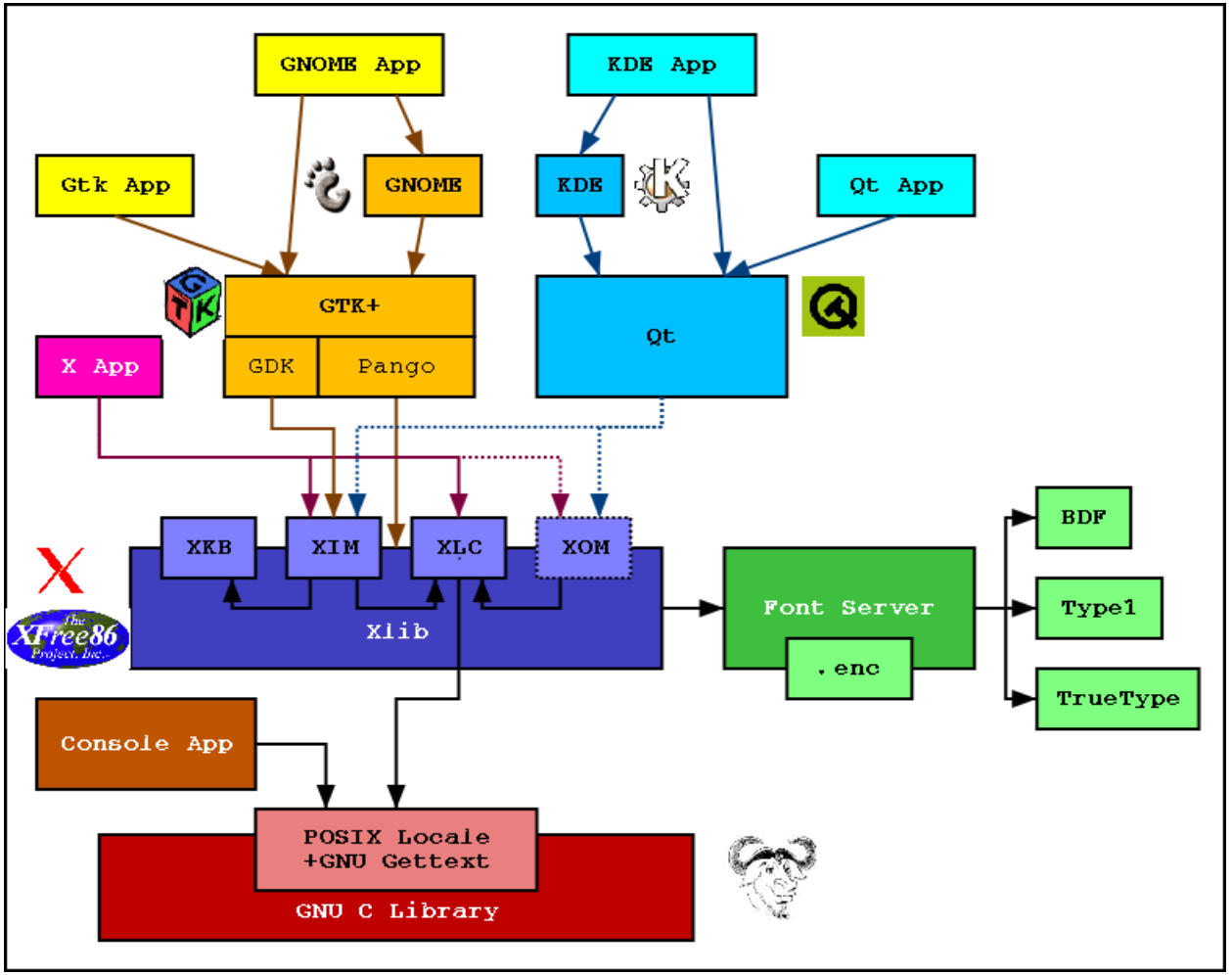
- GNU/Linux 的 i18n/L10n 架構
- 語系 (Locale) 的概念
- 字元集 (charset) 與編碼 (encoding)
- 資訊處理
  - 輸入：輸入法
  - 輸出：顯示、字型、印表
- X Window System 的 i18n 處理
- 存在的問題

GNU/Linux 的 i18n/L10n 架構 (1)

- 觀念釐清
  - 「中文化」是模糊的概念
  - Linux 上的「中文化」兼具 i18n 與 L10n
  - 中文平台發展策略
    - 不需修改既有應用程式，便可顯示、印列、輸入中文
- 不侷限於特定環境
  - Linux tty (console) / Framebuffer
  - X Window System / Toolkit
  - Programmer Editor (Emacs)
  - Embedded / Mobile
  - Cloud

GNU/Linux 的 i18n/L10n 架構 (2)

- 關連鍊 (chain)
  - Linux kernel
  - GNU C Library (glibc)
  - Xorg (X11)
  - Gtk/GNOME
  - Qt/KDE
  - Mozilla / OpenOffice
  - 其他函式庫或工具程式



因為複雜的網頁排版效果考量， Mozilla 自身有一套多國語文輸出處理機制，不全然對應到 Gtk+ 或 Pango，OpenOffice 也是類似的情況

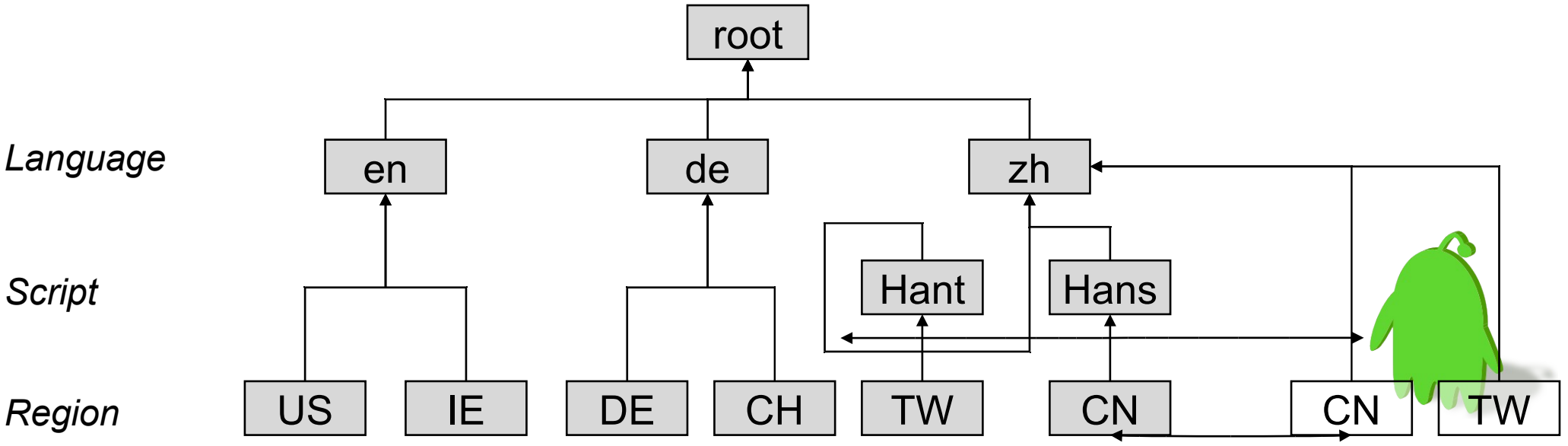
語系 (Locale) 的概念 (1)

- 國際化的四大等級
  - 語言可切換，在系統啟動時可指定某語言
  - 使用不同語言的軟體可同時使用，在應用軟體啟動時可指定某語言
  - **使用不同語言的軟體可同時使用，而且應用程式的語言可動態切換**
  - 使用不同語言的軟體可同時使用，而且在應用程式中可同時使用不同語言

- ✓ mltalk (X 基金會) 決定採納
- ✓ 考量：實用與可行性
- ✓ 原則：不需要重新編譯程式
- ✓ 即可適用多語文環境的平台

語系 (Locale) 的概念 (2)

- 中文語系
  - zh_TW.Big5
  - zh_CN.GB18030
  - zh_HK.Big5-HKSCS
  - zh_TW.UTF-8
- 格式： *language_territory.encoding@modifier*
  - de_AT.ISO-8859-1@euro



語系 (Locale) 的概念 (3)

- LC_COLLATE (比較和排序)

  漢字排序的方式有許多種 , 按照發音 (拼音) 或漢字筆畫來排序是比較容易被接受的方式

- LC_CTYPE (字元分類)
- LC_MONETORY (貨幣單位)
- LC_NUMERIC (數字顯示格式)

  美國： $1,234.56

  德國： 1.234,56DM

- LC_TIME (時間和日期)
  - 12 小時或 24 小時制，時和分之間可用逗點或者冒號隔開
  - [華語] 14 點 20 分， 2000 年三月十四號
  - [英國] 02:20pm 14/03/2000
- LC_MESSAGES (i18n 訊息表示)

區域化環境類別 (categories)

LC_ALL: 此類別可一次設定左列所有的類別
LANG: 作用類似 LC_ALL

語系 (Locale) 的概念 (4)

- 姓名、地址等特殊訊息

  姓名中的「姓」和「名」的先後順序、地址書寫的順序

- 圖示 (Icon) 的通用性

  Icon 設計時需要考慮地區性習慣，及圖形文字的翻譯

- 聲音使用

  不當的聲音或提示會引起反感，此外，聲音的性別對於某些國家是敏感的，如回教國家

- 色調使用：與民俗 / 文化有關
  - 紅色在美國表示危險
  - 紅色在中國表示喜慶

語系 (Locale) 的概念 (5)

- 使用者介面：輸入部份
  - 不同語文或區域，會有不同的 keyboard layout
  - 許多語文會有其特殊的按鍵設計

    如日文的 Kanji/Canna 鍵 (切換漢字 / 假字)，導致軟體無法直接跨越硬體限制
- 文化差異

Trash Can in Thailand

U.S. Trash Can

- 使用者介面：控制元件 (UI controls) 會因不同語文而有差異
  - 區域性訊息
  - 控制元件需有能力動態調整 (察覺語言差異)

字元集 (charset) 與編碼 (encoding)(1)

- CJK scripts use Hanzi ***ideographs*** (表意文字), which totals probably over 100,000 unique characters, with about 10,000 in common use.

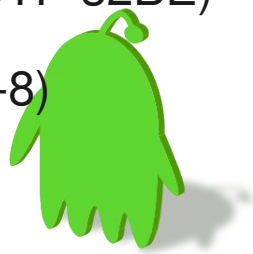
- **Character set:** 給定字元 (character) 的集合
  - {0, 1, 2, 3, ..., 9, A, B, C, ..., Z, a, b, c, ..., z, …}
  - 隨國家與地域的不同，會建立專屬的 character set ，用以區隔地域性

- **Encoding:** 以機器或資料傳輸的角度，字元的位元組表示方式 (可能與 endian 有關)

「學」 (Unicode U+5B78) 可表示為

- “\x5B\x78” (UTF-16BE)
- “\x78\x5B” (UTF-16LE)
- “\x00\x00\x5B\x78” (UTF-32BE)
- “\xE5\xAD\xB8” (UTF-8)

U+5B78 這個表示「學」的字序是固定的，但其表示法會依據 encoding 的設計而不同



字元集 (charset) 與編碼 (encoding)(2)

- 最常見的字元集是 ISO 8859 系列，包含 10 種單一字元多國語言的 encoding / charset
- ISO 8859-1 (Latin1)

  French(fr), Spanish (es), Catalan(ca), Basque (eu), Portuguese (pt), Italian (it), Albanian (sq), Rhaeto-Romanic (rm), Dutch (nl), German (de), Danish (da), Swedish(sv), Norwegian (no), Finnish (fi), Faroese (fo), Icelandic (is), Irish(ga), Scottish (gd), English (en), Afrikaans (af) 和 Swahili (sw) ← **大多數的歐洲語言**
- ISO 8859-2 (Latin2)

  Czech (cs), Hungarian (hu), Polish(pl), Romanian (ro), Croatian (hr), Slovak (sk), Slovenian (sl), Sorbian ← **中東歐語文**
- ISO 8859-3 (Latin3)

  Esperanto (eo) 與 Maltese (mt)
- ISO 8859-4 (Latin4)

  Latvian (lv), Lithuanian(lt), Greenlandic (kl) ,

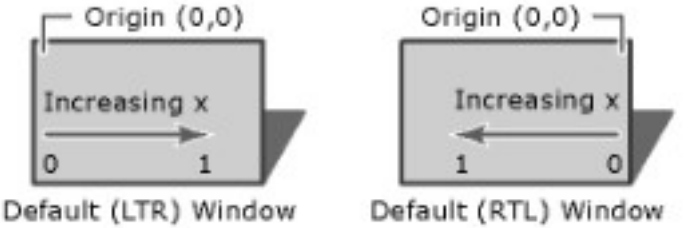
  LappishBulgarian (bg), Byelorussian (be)

字元集 (charset) 與編碼 (encoding)(3)

- ISO 8859-5
  Macedonian (mk), Russian (ru)
  Serbian (sr)
- ISO 8859-6 (阿拉伯語)
  阿拉伯語 (ar)
- ISO 8859-7 (希臘語)
  希臘語 (el)
- ISO 8859-8 (希伯來語)
  Hebrew (iw) 和 Yiddish (ji)
- ISO 8859-9 (Latin5)
  重排 Latin1 ，用土耳其語的若干字母做替換
- ISO 8859-9 (Latin6)
  重排 Latin4 ，去掉了某些符號
- ISO 8859-11 (泰語)
  泰語 (th)
- ISO 8859-12 (Celtic)
- ISO 8859-13 (Latin7)
  Baltic Rim 和 Lativian(lv)
- ISO 8859-14 (Latin8)
  Gaelic 和 Welsh (cy)
- ISO 8859-15 (Latin9)
  Latin1 的變種，修改某些字母

字元集 (charset) 與編碼 (encoding)(4)

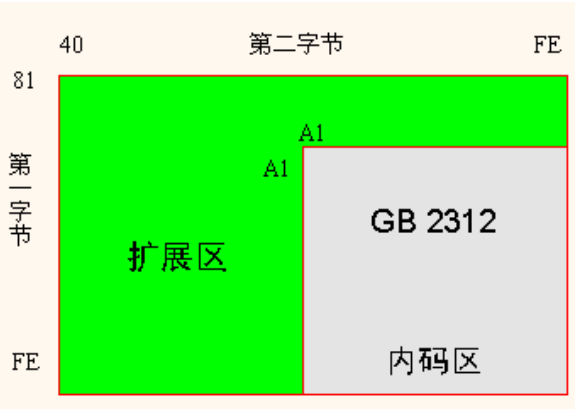
- 多字位元集由 Lead Byte 和 Trail Byte 構成
- 在 CJK 的處理來說，一個字元佔用了至少兩個 byte，在應用程式處理增加了困難度
  - 裝置顯示上，游標的位置需要依據字元而定
    - 而非 byte 數
  - 刪除和移動時必須是整字操作
    - 而且有方向性的議題 (Left-To-Right / Right-To-Left Writing System)，如希伯來文與阿拉伯文
  - 輸入上需要輔助性的 preedit(預選字詞) 處理



字元集 (charset) 與編碼 (encoding)(5)

Charsets in Mainland China

- GB 2312-80 (國家標準):
  - 0x2121-0x7E7E or 0xA1A1-0xFEFE
  - 6763 Hanzi (簡體中文加符號)
- GB 12345-90: 繁體中文
- GBK (1995) (規格):
  - {0x81-0xFE}{0x81-0xFE}
  - 21003 CJK 漢字 (包含在 Unicode 2.1 規範的 20920 漢字) ，也就是 Microsoft Windows 的 CP936
- GB 18030-2000 (國家標準)



字元集 (charset) 與編碼 (encoding)(6)

Charsets in Taiwan (1)

- CNS 11643-1986, -1992 (國家標準)
  - 94 x 94 x 14 planes
  - 因缺乏商業實作支援，多限定於政府機關
- Big-5 (產業標準規格):
  - Codespace: {0x81-0xFE}{0x40-0x7E,0xA1-0xFE}
  - Big-5 (1984): 原始規格，約 13053 個繁體中文漢字

    Big-5 (ETen): 新增日文假字
  - Big-5 (CP950): 在 Microsoft Windows 採用

字元集 (charset) 與編碼 (encoding)(7)

Charsets in Taiwan (2)

- Big-5 變體
  - Big-5+: 相當於 GBK 的 Big5 版本 (20000+ 字)
    - Codespace: {0x81-0xFE}{0x40-0x7E,0x80-0xFE}
    - 被台灣教育部所拒絕
    - 包含簡體中文，對使用者造字區的缺乏
  - Big-5E: 較少的突破，只使用既有的使用者造字區
    - 新增 4000 字
- CCCII (主要用於台灣圖書館系統)
  - 採用 3-byte encoding

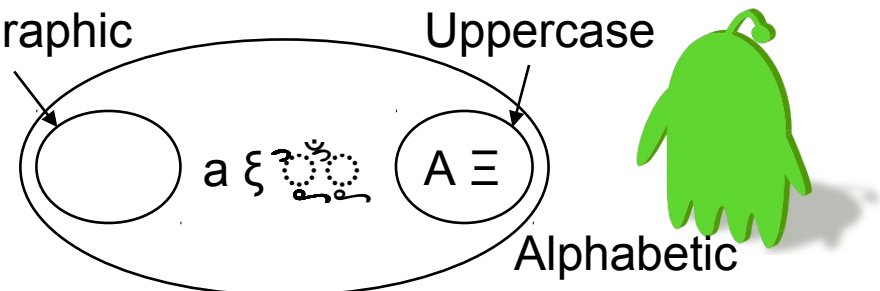
字元集 (charset) 與編碼 (encoding)(8)

ISO 10646 與 Unicode

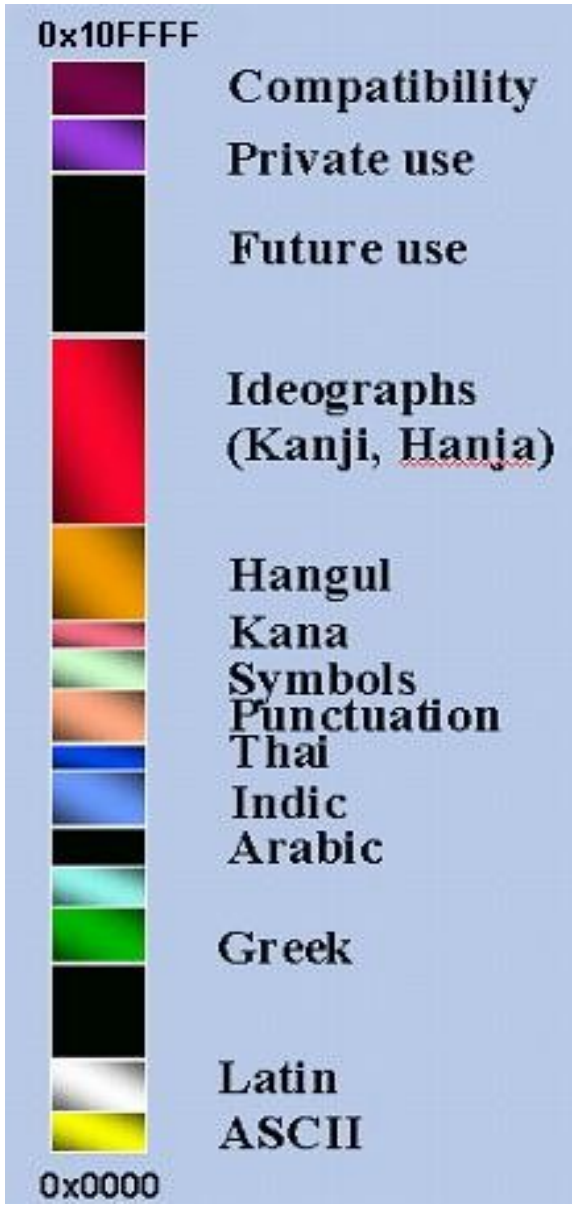
- 由非營利組織 Unicode 研討會維護和改進
  - 起源於 Xerox 和 Apple 之間的合作研究
    若干公司組成了一個非正式的論壇
  - 接著 IBM 、 Microsoft 等公司迅速加入
- Unicode 研討會在 1990 年發表 Unicode 第一版
  - 同一時期 ISO 完成類似的編碼，也就是 ISO 10646

考量沒必要有兩套標準， Unicode 研討會和 ISO 在 1991-1992 年間整合

- Unicode 包含廣泛使用的字元
  - 如世界通行的書寫語言、印刷用字、數字、科學符號
    地理圖形，以及標點符號
  - 甚至包含「顏文字」



Unicode Character Set



Example Unicode Characters

ASCII	ABCDEFGHIJKLMNOP
Latin-1	ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏ
Latin-2	āĂăĄąĆćĈĉĊċČčĎďĐ
Greek	ΫABΓΔEZHΘIKΛMNΞO
Cyrillic	рстуфхцчшщъыьэюя
Thai	ภมยรฤลฦวศษสหฬอฮฯ
CJK	北両丢乖两严並丧
Korean	감갑값갓갔강갖갗

Unicode 的組合性

Base + non-spacing (combining) character(s)

A + ˚ = Å

U+0041 + U+030A = U+00C5

a + ˆ + . = ậ

U+0061 + U+0302 + U+0323 = U+1EAD

a + . + ˆ = ậ

U+0061 + U+0323 + U+0302 = U+1EAD

- 注意： Unicode 表示法為 U+hhhh

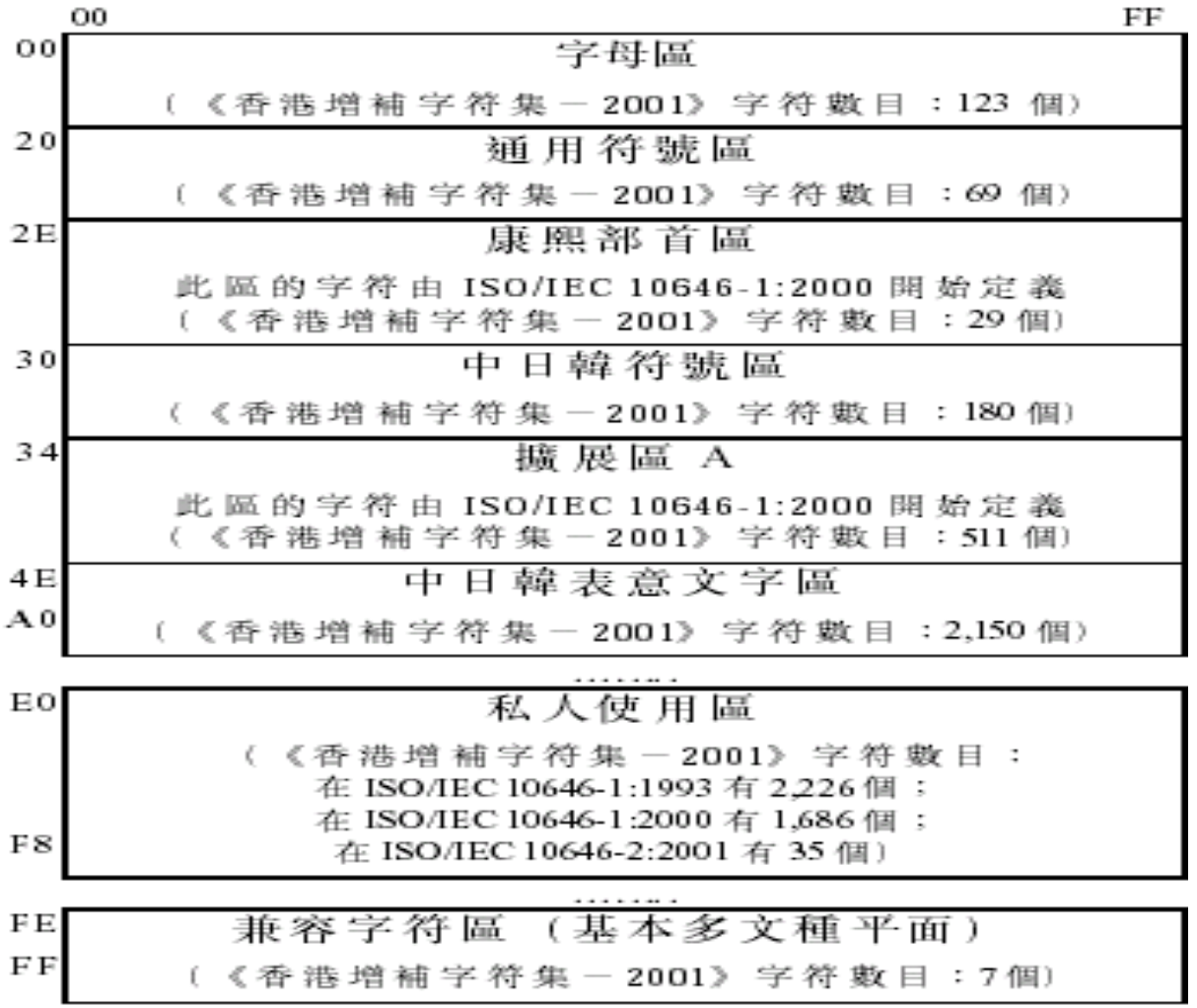
字元集 (charset) 與編碼 (encoding)(9)

ISO 10646 與 Unicode

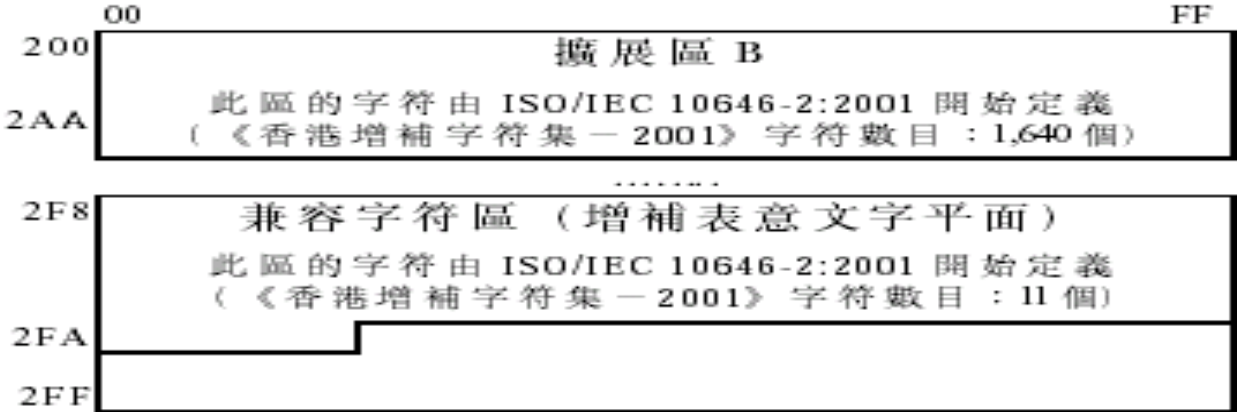
- Unicode 作為一種編碼也有缺陷
  - 字元編碼的位置與排序無關

    故，軟體支援 Unicode ，僅是國際化的第一步
  - 實際應用還與語言相關的訊息和規則有關
  - 須提供與其他編碼的雙向轉換表格
- 雖使用 Unicode 會使普通的英文文本大兩倍，但使用 Unicode 的整個系統卻不會增加太大
  - 考量到系統存放的文件多為二進位文件格式
  - 使用針對 Unicode 的壓縮方式，可把文件壓縮成和使用對應的 8-bit 文字一樣大小

基本多文種平面



增補表意文字平面



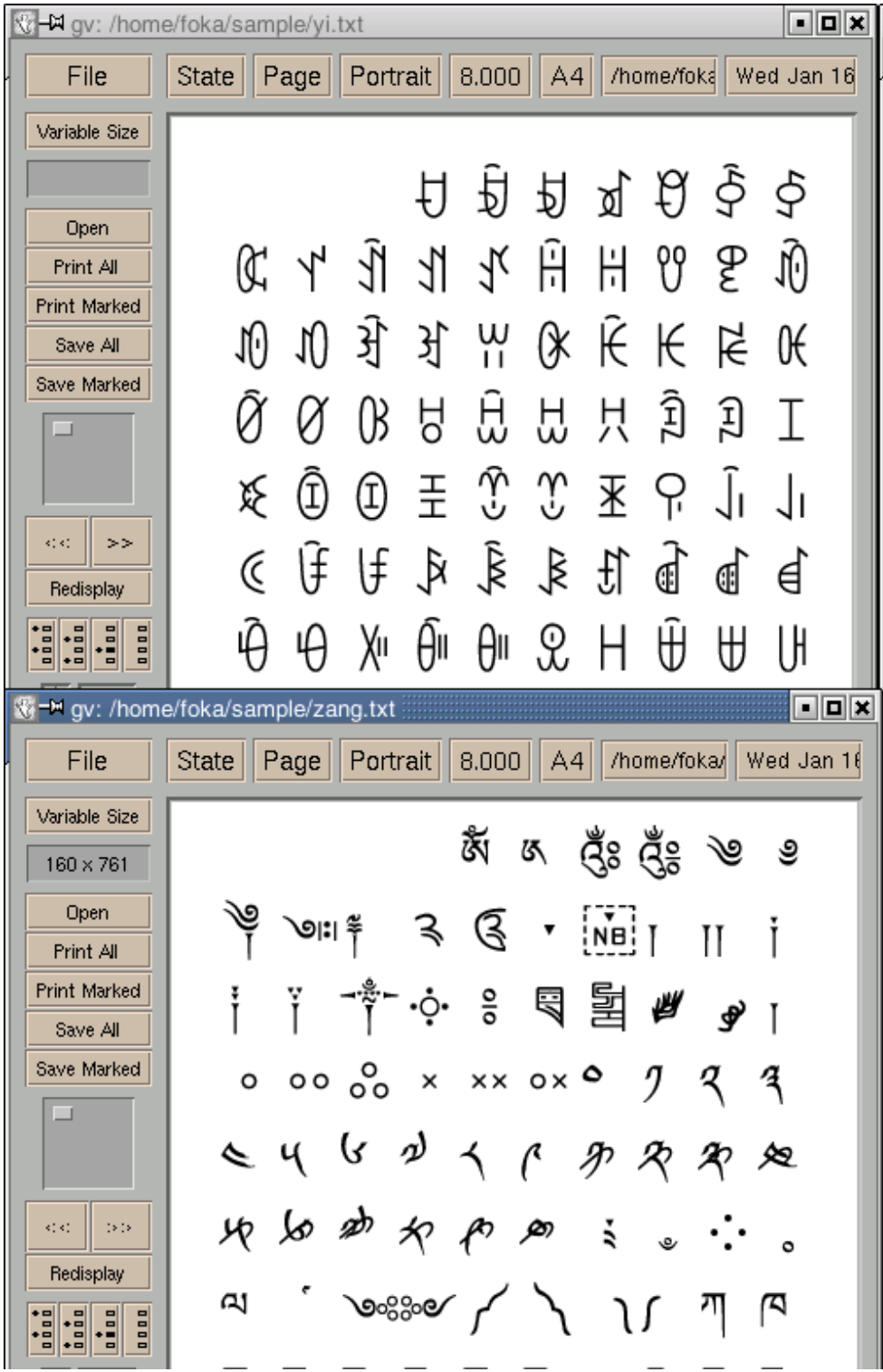
12/06/02

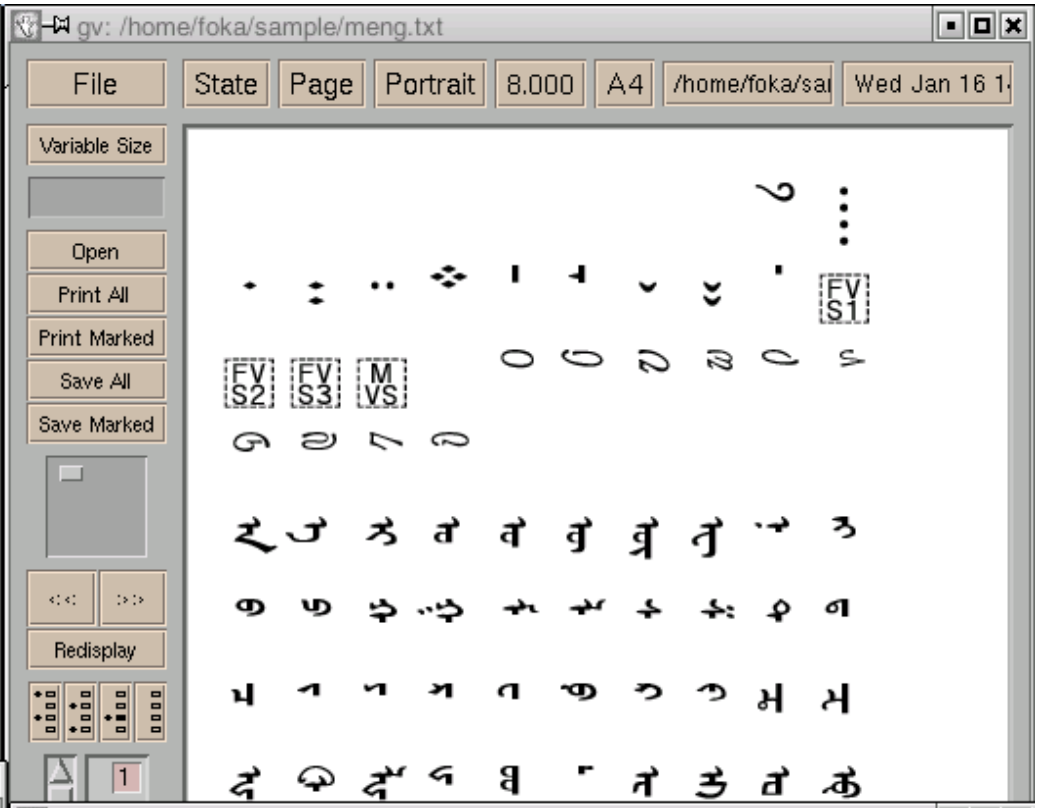
CJK Extension C1 V2.1_E1

C1 V2.0 C1 Num KangXi Radical Strk Fs	G Source H Source M Source T Source J Source	K Source Kp Source S Source V Source U Source	Image
00001			
00001	HK-8840		
0075.011			
—			
0 | 0			
00002			
00002			
0076.021			
—	TC-2121		
1 | 5			
00003		K5H00574	
00003			
0076.021			
—			
1 | 5			
00004			
00004			
0076.141			
—		V04-4021	
2 | 2			
00005	BK100001		
00005			
0076.141			
—			
2 | 2			
00006			
00007			
0076.141			
—	TC-2143		
2 | 3			
00007	CYY00001		
00008			
0076.141			
—	TC-212E		
2 | 3			
00008	CYY00002		
00009			
0076.141			
—	TC-2133		
2 | 3			
00009		K5H00575	
00010			

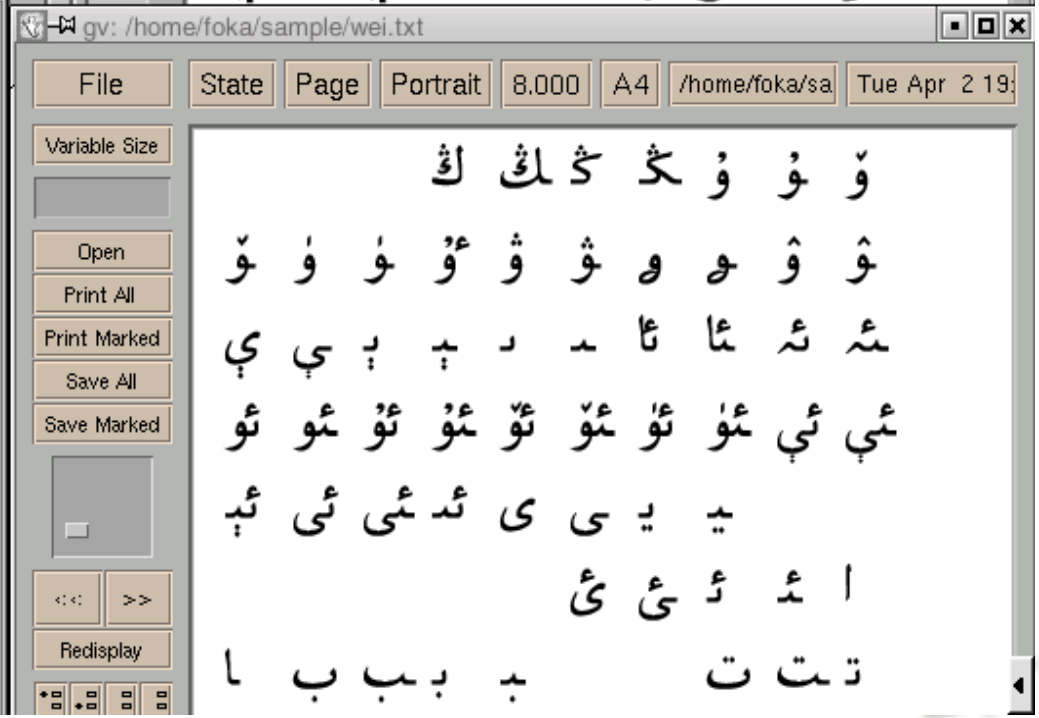
C1 V2.0 C1 Num KangXi Radical Strk Fs	G Source H Source M Source T Source J Source	K Source Kp Source S Source V Source U Source	Image
00014		K5H00579	
00017			
0076.141			
—			
2 | 5			
00015		K5H00584	
00019			
0077.061			
—			
3 | 1			
00016		K5H00589	
00020			
0077.061			
—			
3 | 1			
00017		K5H00590	
00021			
0077.061			
—			
3 | 1			
00018		K5H00585	
00022			
0077.061			
—			
3 | 2			
00019		K5H00587	
00023			
0077.061			
—			
3 | 2			
00020		K5H00588	
00024			
0077.061			
—			
3 | 2			
00021		K5H00591	
00025			
0077.061			
—			
3 | 2			
00022		K5H00592	
00026			

C1 V2.0 C1 Num KangXi Radical Strk Fs	G Source H Source M Source T Source J Source	K Source Kp Source S Source V Source U Source	Image
00027	BK100002		
00035			
0077.061			
—			
3 | 5			
00028		K5H00581	
00036			
0077.061			
—			
3 | 5			
00029		K5H00583	
00037			
0077.061			
—			
3 | 5			
00030		K5H00594	
00038			
0077.061			
—			
3 | 5			
00031		K5H00603	
00041			
0078.021			
—			
4 | 1			
00032		K5H00609	
00042			
0078.021			
—			
4 | 1			
00033			
00043			
0078.021			
—		V04-4022	
4 | 2			
00034		K5H00601	
00044			
0078.021			
—			
4 | 2			
00035		K5H00602	
00045			


gv: /home/foka/sample/yi.txt
File
State
Page
Portrait
8.000
A4
/home/foka
Wed Jan 16
Variable Size
Open
Print All
Print Marked
Save All
Save Marked
Redisplay


gv: /home/foka/sample/meng.txt
File
State
Page
Portrait
8.000
A4
/home/foka/sa
Wed Jan 16 1
Variable Size
Open
Print All
Print Marked
Save All
Save Marked
Redisplay
FV S1
FV S2
FV S3
M VS

gv: /home/foka/sample/zang.txt
File
State
Page
Portrait
8.000
A4
/home/foka/
Wed Jan 16
Variable Size
160 x 761
Open
Print All
Print Marked
Save All
Save Marked
Redisplay
NB


gv: /home/foka/sample/wei.txt
File
State
Page
Portrait
8.000
A4
/home/foka/sa
Tue Apr 2 19:
Variable Size
Open
Print All
Print Marked
Save All
Save Marked
Redisplay

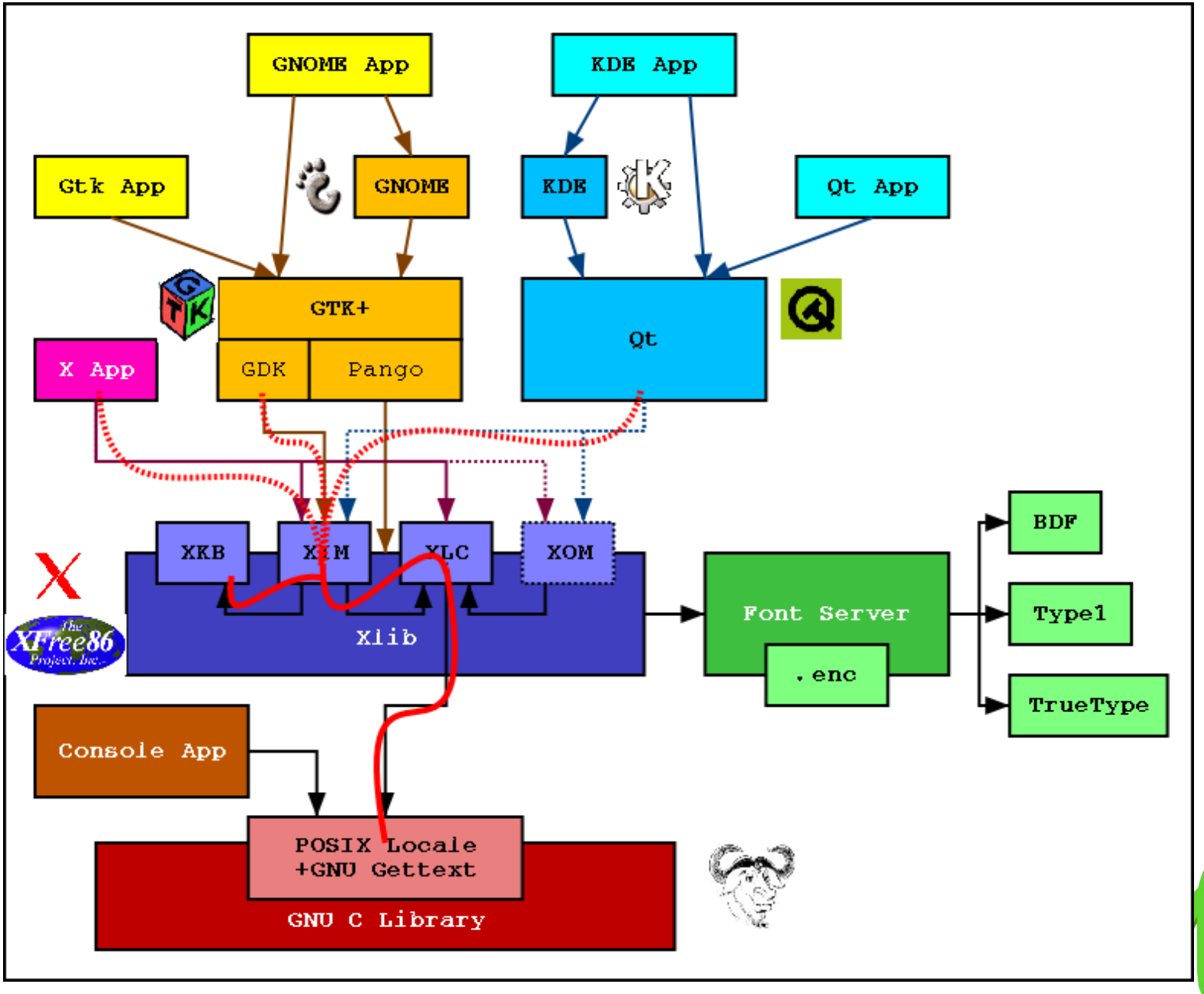
哀悼樋浦秀樹 (Hideki Hiura) 先生

- 因癌症逝世於美國 (Apr 7, 2010)

  "Hideki Hiura is chief scientist and CTO of JustSystems, Inc. He is a founder and chairperson of **OpenI18N.org/Free Standards Group**, an independent, nonprofit organization dedicated to accelerating the use of free and open source software by developing and promoting standards. He is also a founding member of **W3C I18N WG**. As an architect at Sun Microsystems, he was involved with variety of standards and standard organizations including ISO, W3C, OMG, The Open Group, OSF, Unix International, X Consortium and Unicode."

- 整合過去在 Sun Microsystems 的成果，透過 OpenI18N 與 Free Standards Group 等組織，推廣到開放的平台，為今日的資訊系統做出了巨量的貢獻

GNU/Linux 的 i18n 輸入架構



以中文輸入為例

- 以「字型」為主
  - Cangjie (Changjei) (倉頡)
  - Wubizixing (五笔字型)
  - Array30 (行列)
- 以「字音」為主
  - Pinyin (拼音)
  - Zhuyin (Phonetic) (注音)
  - Cantonese Pinyin (廣東話拼音、粵拼)
- 以「字義」為主
  - 辭典輸入法 (英漢、漢英)
- 混合「形、音、義」
  - 如「智能拼音」

早期台灣的資訊系統對拼音不是很講究，
所以才會有 Cangjie 這類拼法

輸入法技術

- XIM (X Input Method) 回顧
- 走出 XIM ，迎向輸入法新架構
  - IIIMF
  - UIM
  - SCIM
  - ibus

XIM 特徵

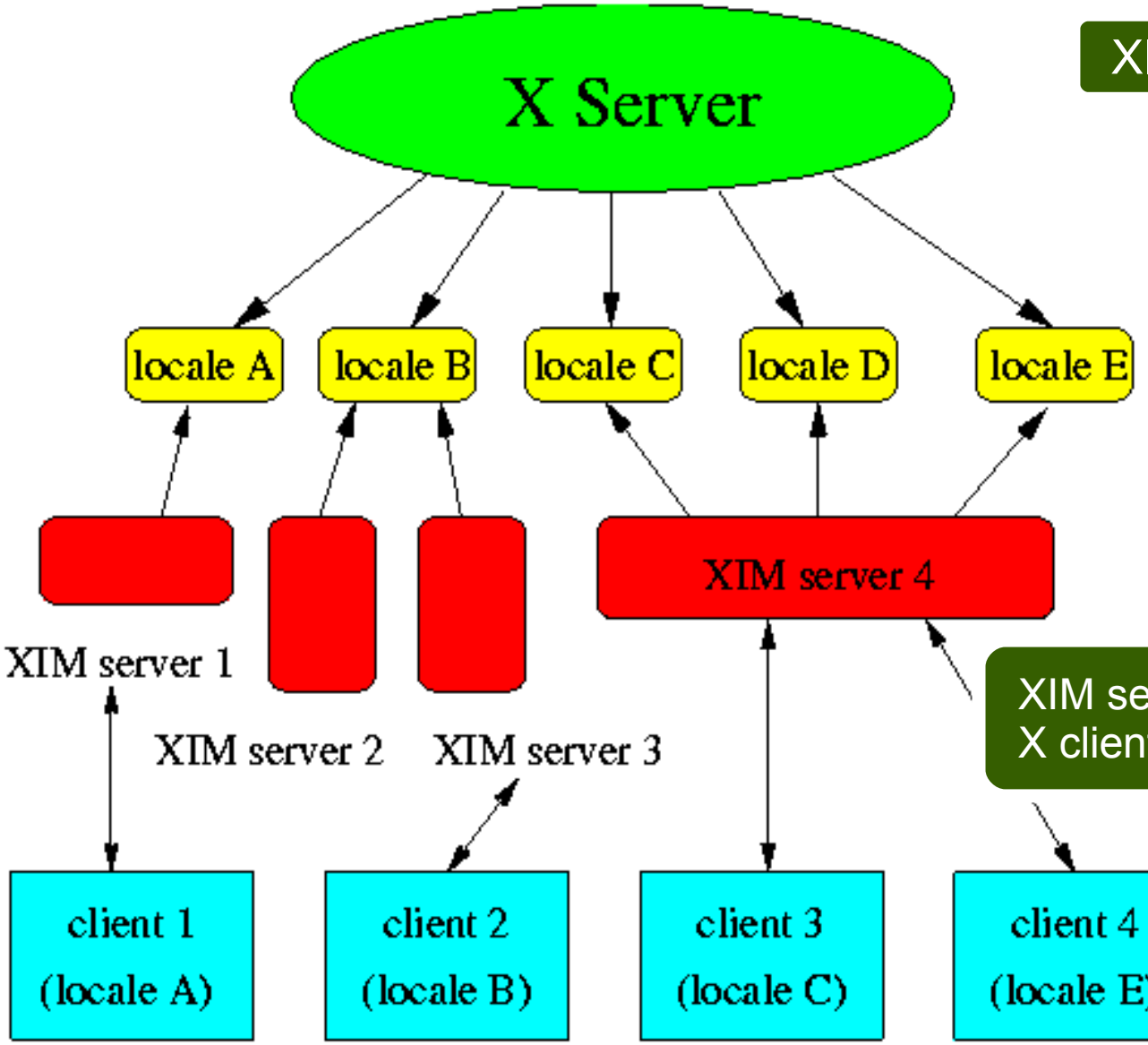
- 架構於 I18N 與 Locale 上的 X 輸入法協定
- 目的：不更改原始碼本身，即可接受各語系輸入法
- 用來處理非歐美語系的輸入法
- 規範
  - XIM Client： 一般 X 應用程式
  - XIM Server： 輸入法

- X11R4
  - 沒有統一的輸入法系統，由個別軟體自行處理輸入法需求
  - 由 Fujitus 與 Omron 提供 Xjp
- X11R5
  - 1989 年 Xi18n 開始在 X 協會推動輸入法
  - 開始定義 XIM / XFontSet APIs
  - 分歧發展
    - Omron/NTT 實作 Xsi
    - Unix International (UI) 實作 Ximp
    - Unix International 將 XIM 定義為標準

- X11R5
  - 輸入法系統的衝突
    - Omron/NTT
    - UI (Fijitsu、 Sony、 Xerox、 ATT/USL、 Hitachi、 Sun、 ...)
- X11R6
  - OSF (IBM、 DEC、 HP) 加盟 UI
  - XIM Protocol 定義（但不與 Xsi/Ximp 相容）
  - 強化 XIM / XOM / XLC APIs
  - Xlib XIM 實作獨立為 SI
  - IMdKit (IM server Developer Kit) 引入

- X11R6.5
  - X 協會重新啟動， X.org 更新 X 標準與 SI
  - Sun 移轉 Solaris 的 Xi18N 貢獻到 X.org
  - Xfree86 從 Xi18N 相關改變帶回 X.org
  - OpenI18N.org 發佈 xiiimp.so
    - XIM 相容 IIIMF 模組

XIM 架構



XIM 與 X server 無關

XIM server 實際上也是一種 X client，與 X server 溝通

X 應用程式依據 locale，連結上預先註冊上某 (些) locae 的 XIM server

XIM 的缺陷

案例： Firefox 可進行多國語文網頁瀏覽，可是卻缺乏對應的 XIM client/server 互動

- XIM 結構與 X Window System 緊密相連

  對 X11 元件具有強烈的相依性：
  - XfontSet, Cordinates, Color, Display, Screen, Atom, …
  - Ctext (subset of ISO-2022) 更是對 charset 的依賴
  - 採 X transport ，因此 XIM 只能存在於 X
- 無法有效支援多種語言
  - XIM 以 POSIX 單一 locale 為基礎，導致每個 XIM client/server 聯繫以 locale 為主，無法更動
  - XIM Server 無法告知 Client 端，用戶目前輸入何種語言或編碼

現在 X Toolkit(如 Gtk+ 與 Qt) 的趨勢則是自行實做專屬的 IM (Input Method) module， 以提供充分的使用與開發彈性

忘了 XIM 吧

- XIM 設計者回顧 Hideki Hiura 指出：

  「基本上 XIM 太多問題... 根本無法透過 patch 改正」

- 1995 年重新設計輸入法，提出 IIIMF 架構 **(Internet/Intranet Input Method Framework)**
- IIIMF 是 openi18n 官方的輸入法架構

注意，自從 FreeDesktop/Xorg 的開發導入重新設計的 XCB (X C Binding)，全面取代傳統的 Xlib 後，原本的 XIM 就重新置放於 XCL(Xlib Compatibility Layer) 中，致使新的技術問題發生，且延宕多時才克服。適逢眾多 Linux 套件全面移轉到 SCIM 與 iBus 輸入法架構上

新一代輸入法架構

- 許多輸入法系統相繼提出
- 但唯有兼具質（彈性、擴充性）與量（支援多語文、多輸入法需求）者將勝出
- GNU/Linux distribution 的選擇
  - **IIIMF**

    過去由 Sun Microsystems 主導，OpenI18N 正式計畫
  - **SCIM**

    由 SCIM 與 UIM 作者共同推動，已整合多項輸入法成果
  - **iBus**

    由 RedHat 推動，已整合多項輸入法成果

無可否認的事實， Linux Distribution 對於輸入法的收錄標準，通常會考量到通用性與活躍度

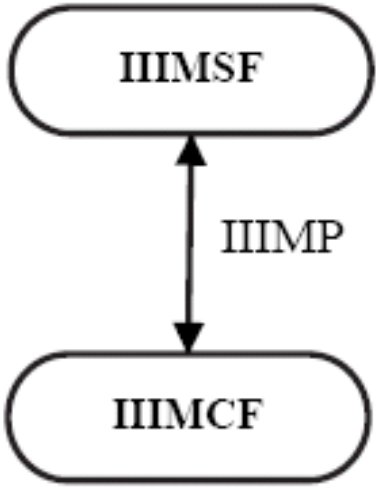
XIM 與 IIIMF 的對比

XIM	IIIMF
應 client 端需求，可能產生多個 Server	Daemon 存在，只需啟動一次
應用程式受 Locale 限制 轉換狀態需更換 Locale 變數	與 Locale 無關 直接切換
Locale 切換較麻煩	不需切換 Locale
輸入法引擎等同於 XIM Server	輸入法引擎以動態模組掛入 IIIM Server
必須有 X Window system	與視窗 / 操作環境無關
	可相容於支援 XIM 之應用程式

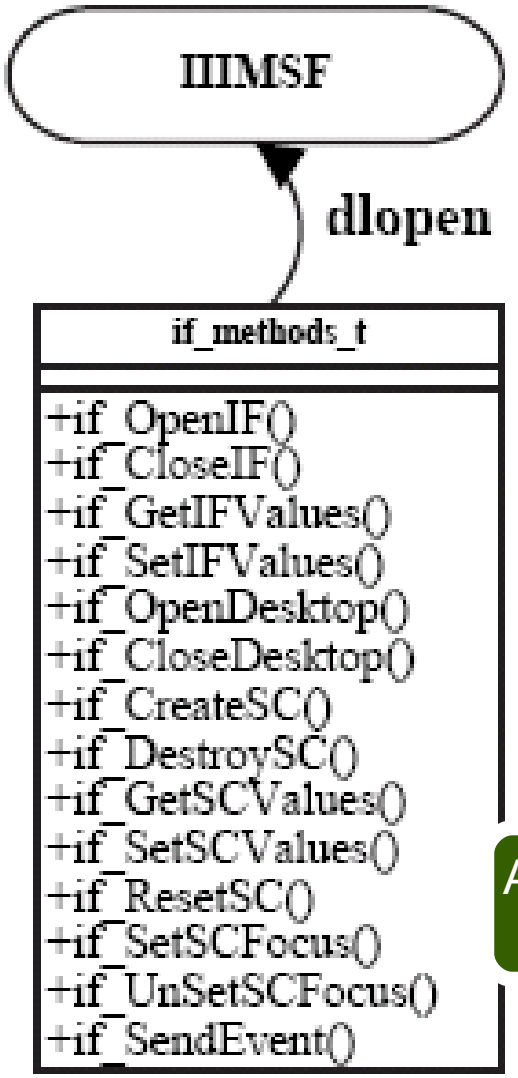
[解釋] IIIMF 相當於抽離傳統 XIM server 的設計，分為平台相關的銜接部份 (IIIM X/Gtk/Qt framework) 與輸入法引擎 (Language engine)，而開發者只要專注於輸入法引擎本身即可

IIIMF 發展目標

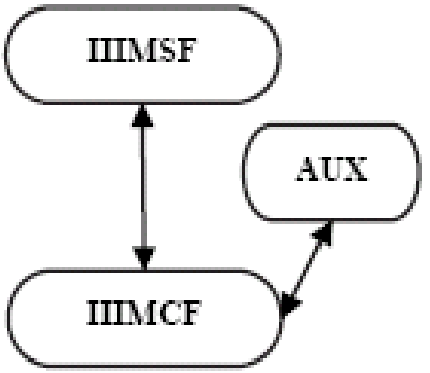
- 許多徹底解決 XIM 弊端
- 去除對 Window System 的相依性
  - 完全 Unicode 支援、實現真正多語文支援
- 完整的 IM 架構，可透過網路動態提供輸入法
  - 在 UNIX 運作的 server 可讓 Windows 環境的 client 使用
- 可攜性與擴展能力
  - 提供現有 XIM, GTK+ im-module, Qt im-module, Java Input Framework 等架構的溝通介面
  - 擴展性強的 client-side API



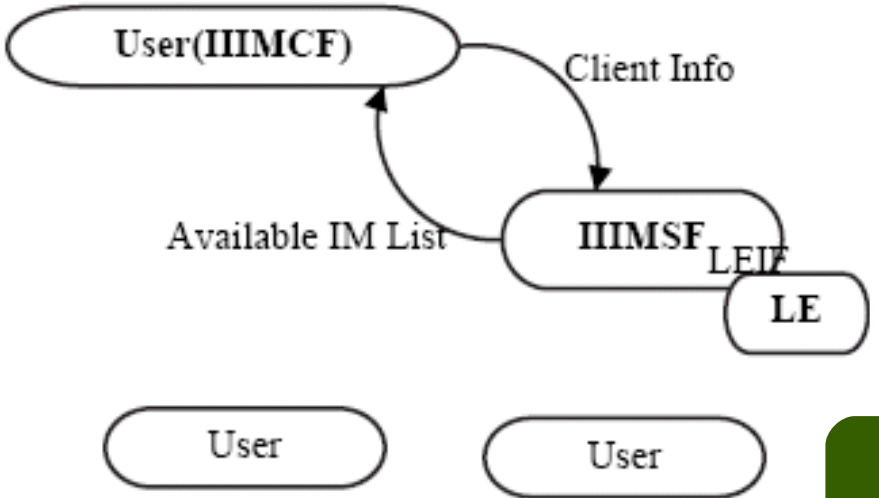
制定全新的 Protocol: IIIMP



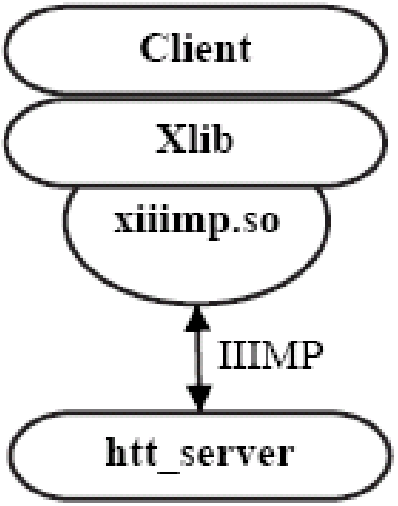
IIIMSF =
IIIM Server Framework



AUX 是配合 Client 的輔助處理機制，比方說候選字詞視窗



LE =
Language Engine，由
IIIM Server 動態載入



銜接傳統 XIM 的方式，只要實做 IIIM X Client Framework

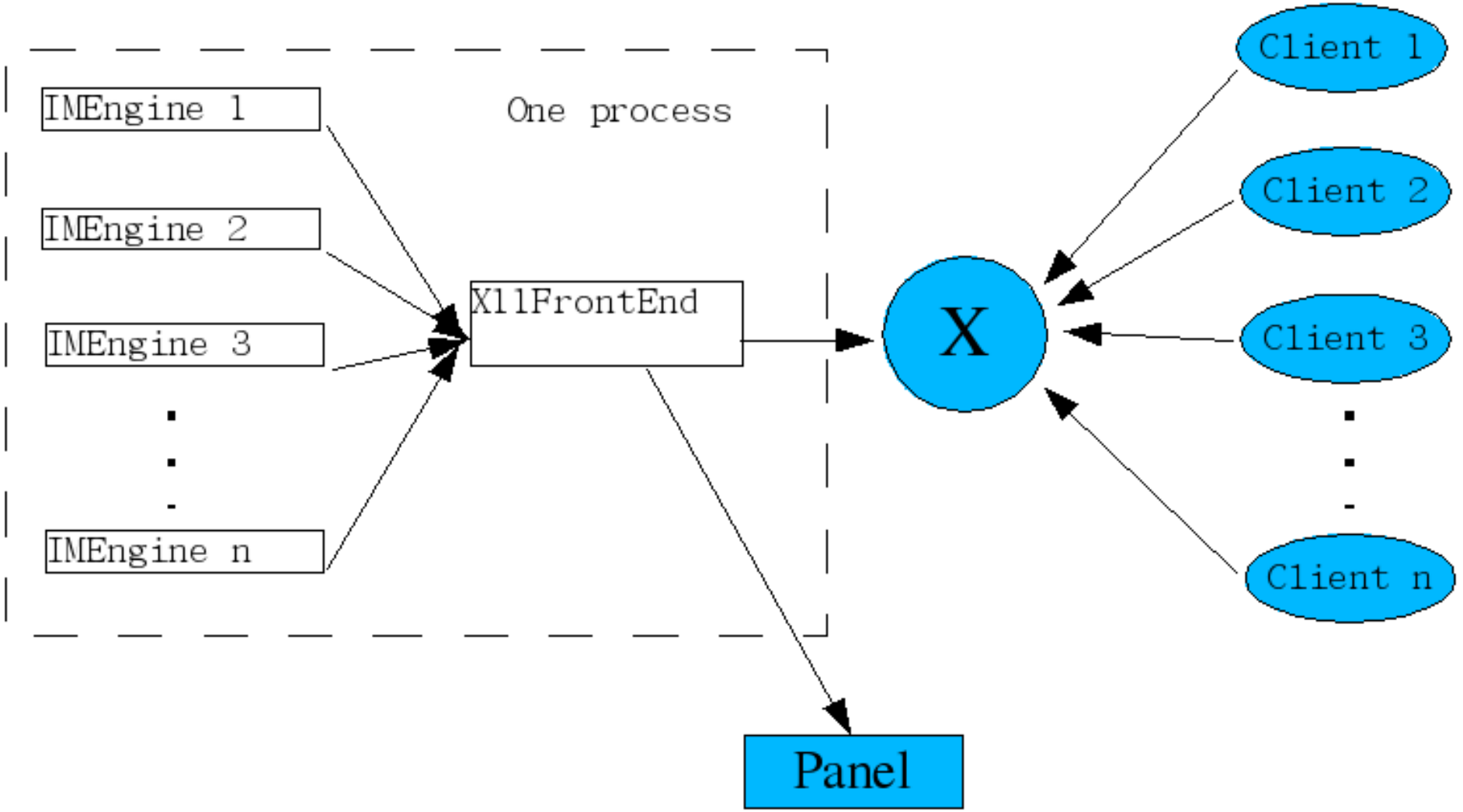
IIIMF 的發展狀態

- 一度聲勢浩大， Fedora Core 2 (2004 年) 與 RedHat Enterprise Linux 4 (2005 年) 皆內建 IIIMF 作為輸入法架構
- 但主要的輸入法引擎仍由 Sun Microsystems 所維護，在 JDS (Java Desktop System) 產品開發時，曾將若干輸入法引擎開放原始碼 (GPL 授權， 2004-2006)
- 2005 年後，因為 IIIMF 在 GNU/Linux 面臨眾多穩定度與可用性的問題，逐漸被新興的 SCIM 輸入法架構取代， Fedora Core 5 開始改為以 SCIM 為預設輸入法平台
- Hideki Hiura 與若干主力工程師離開 Sun Microsystems， IIIMF 開發停滯

SCIM (Smart Common Input Method)

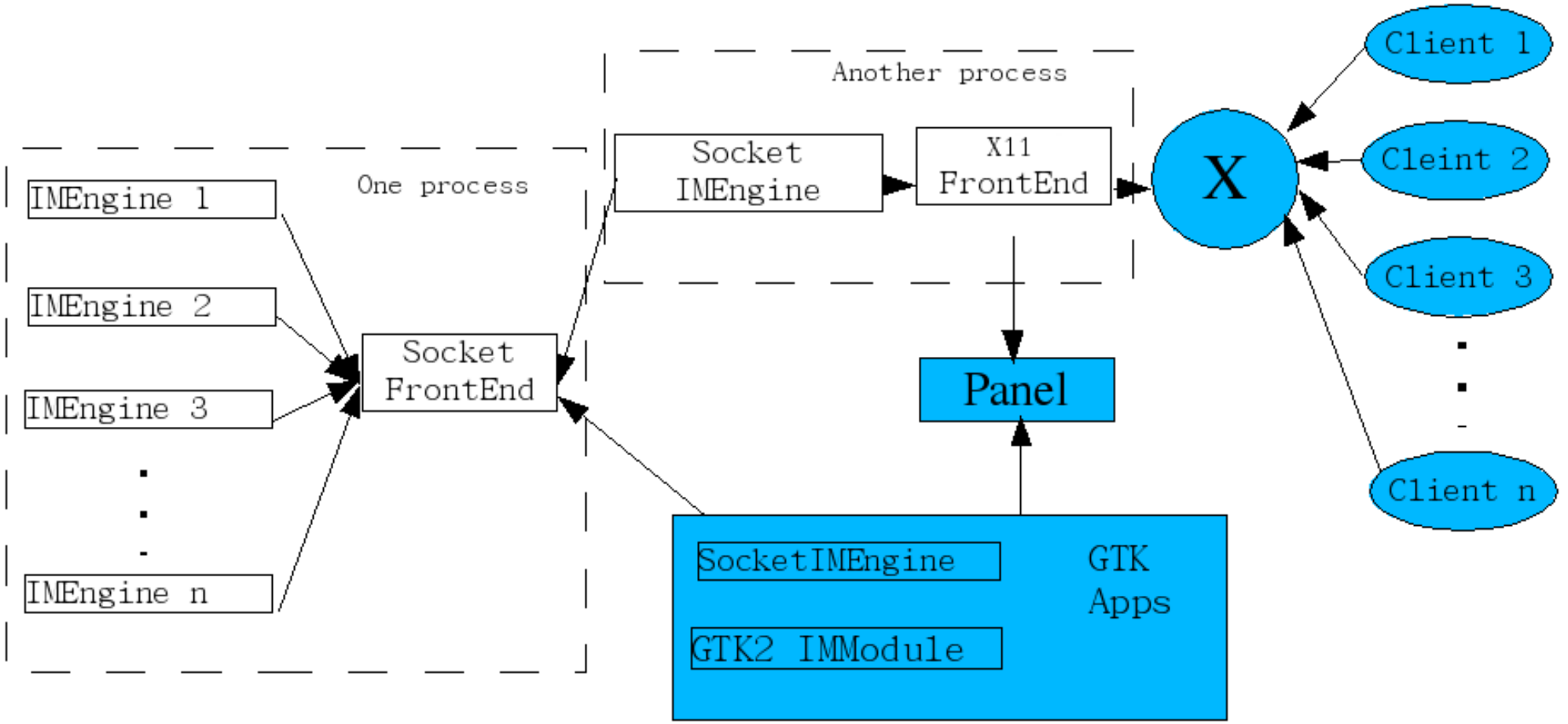
- 發展背景
  - SCIM, UIM 與 m17n library 原本是三路人馬
  - SCIM (Smart Common Input Method platform) 為蘇哲發展，完整的簡體中文架構
  - UIM (Universal Input Method) 為 TOKUNAGA Hiroyuki, Masahito Omote, Yamaken 等人發展，具備眾多 CJK 輸入法支援
  - m17n 則如名稱 multilingualization 所示，廣泛支援多種語文
- SCIM + UIM + m17n: 完整的多國語文輸入解決方案
  - 2004-06-12 SCIM/UIM 整合
  - 2004-06-14 SCIM/m17n 整合

SCIM 運作模式 (1): 動態載入 IMEngine



此機制類似 UIM 的運作模式

SCIM 運作模式 (2): Client-Server 架構



此機制類似 IIIMF 的運作模式

SCIM 的發展狀態

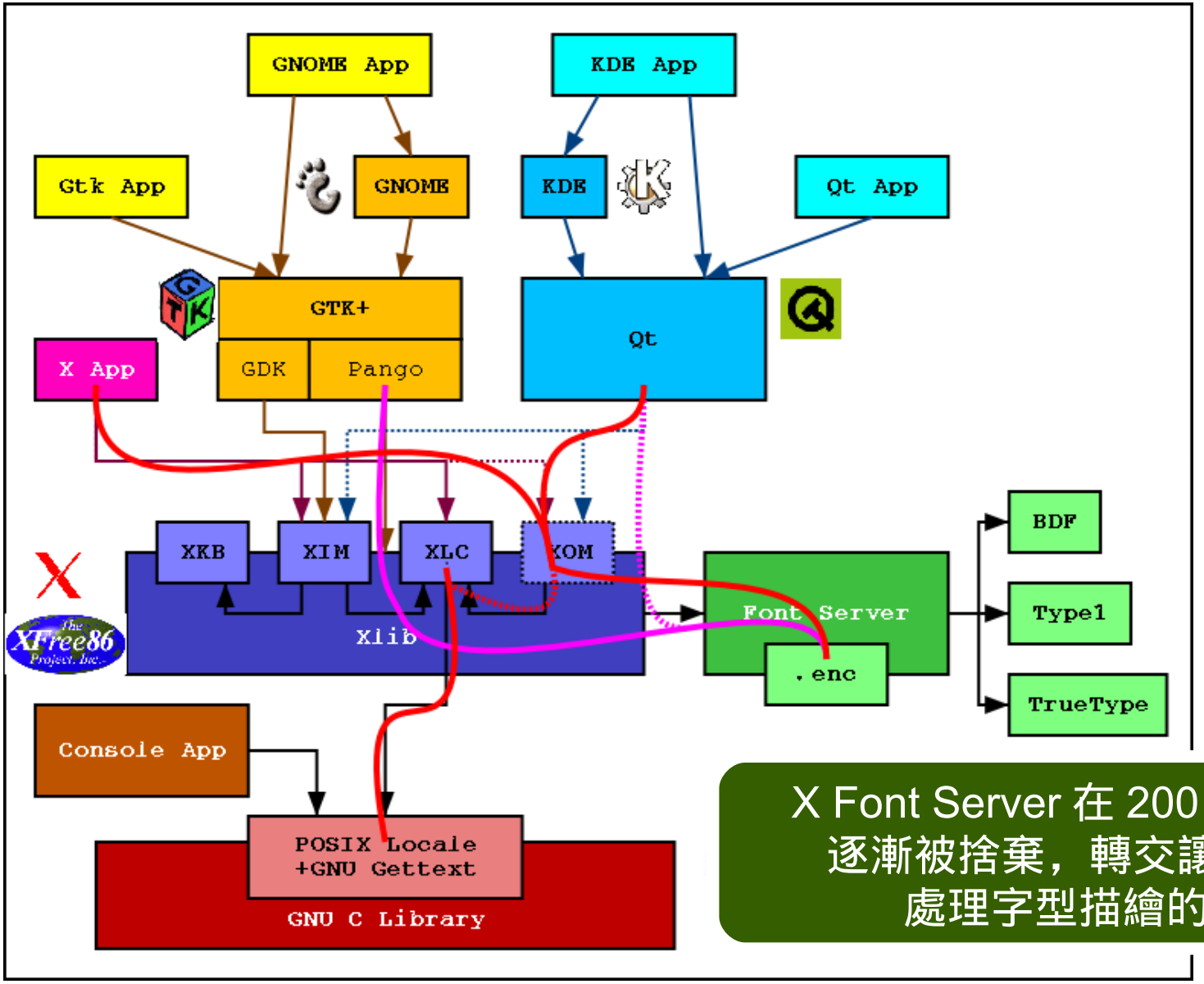
- SCIM 讓 Linux 桌面應用邁入新的里程碑，除了典型的中日韓輸入法外，包含手寫辨識引擎，也整合到 SCIM
- 經過修剪後， SCIM 甚至可在 Nokia N900 手機 (基於 Maemo/Linux 作業系統) 順利運作

iBus (Intelligent Input Bus)

- SCIM 開發停滯
  - IM-Bus (蘇哲)
  - SCIM-2 (胡正)
- 在東北亞 OSS 論壇所提出的「輸入法引擎服務提供者介面規格」(Specification of IM engine Service Provider Interface) 草案裡，能實現以 Bus 為核心的架構，被建議採用
  - 當時服務於 RedHat 的黃鵬參考此規範

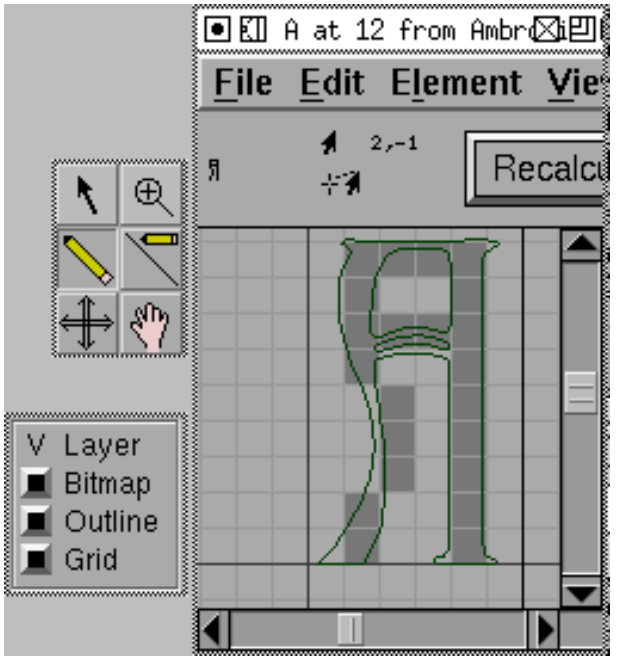
    改用 D-Bus, Glib, Python 等 GNU/Linux 常見的元件重新開發 IM-bus
- 以 C 語言及 Python 語言開發的 iBus 成為最新主流輸入法架構

GNU/Linux 的 i18n 輸出架構



X Font Server 在 2001 年之後，
逐漸被捨棄，轉交讓 X client
處理字型描繪的動作

字型 / 字體 / 字庫



- 點陣字庫 (Bitmap fonts)

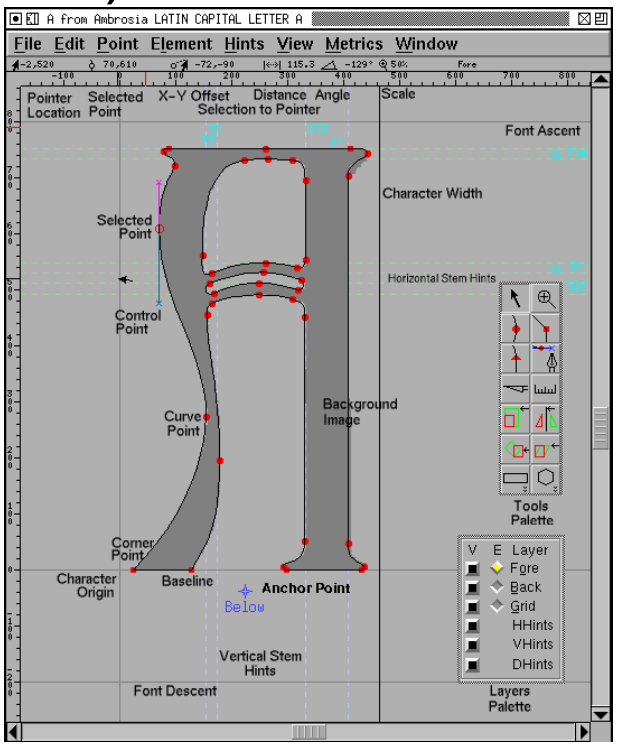
  Console fonts

  BDF/PCF fonts in X Window System

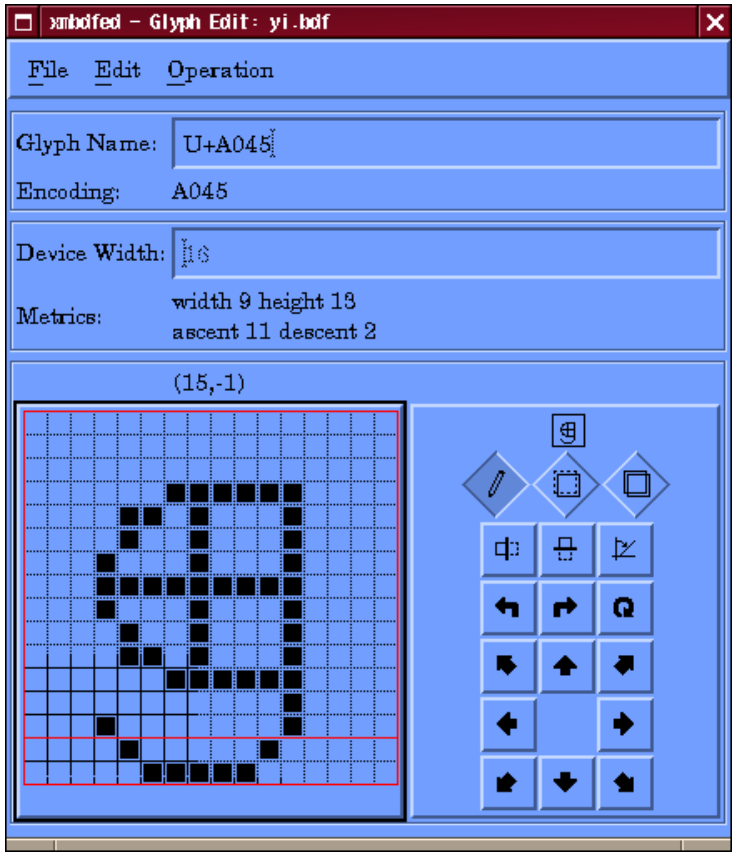
- 向量 / 矢量 / 曲線字庫 (Vector fonts)
  - PostScript Type 1 fonts (Adobe)
  - TrueType (Apple, Microsoft)
  - OpenType fonts

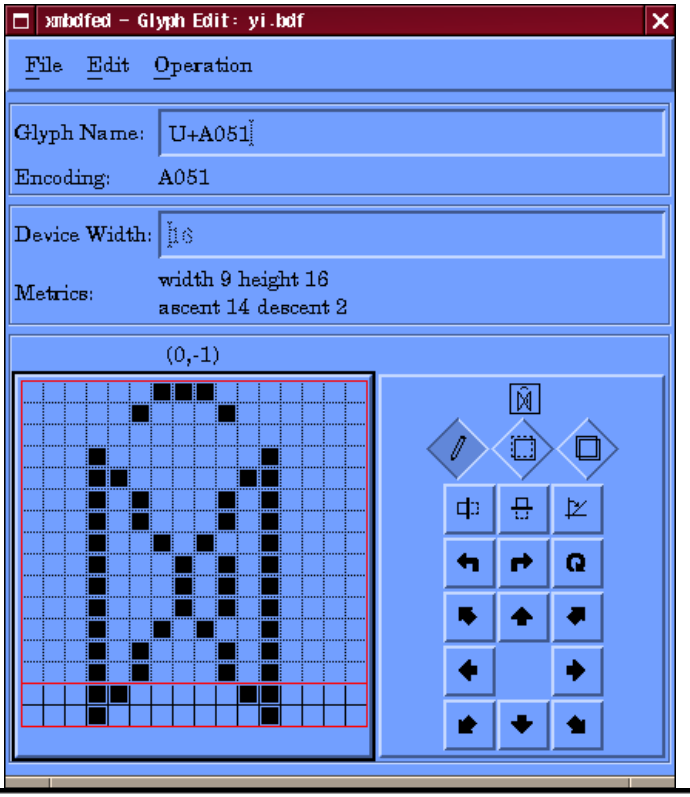
    (可能包含 TrueType 或 Type 1 字型)

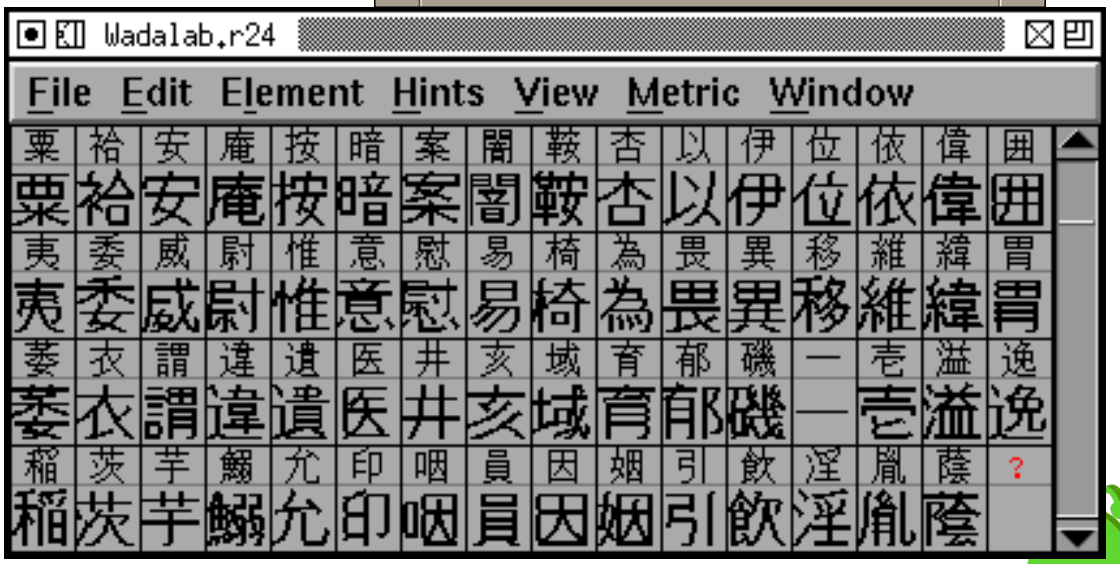
    (Adobe, Apple, Microsoft)



字型 / 字體 / 字庫


xmbdfed - Glyph Edit: yi.bdf
File Edit Operation
Glyph Name: U+A045
Encoding: A045
Device Width:
Metrics: width 9 height 13
ascent 11 descent 2
(15,-1)


xmbdfed - Glyph Edit: yi.bdf
File Edit Operation
Glyph Name: U+A051
Encoding: A051
Device Width:
Metrics: width 9 height 16
ascent 14 descent 2
(0,-1)


Wadalab.r24
File Edit Element Hints View Metric Window

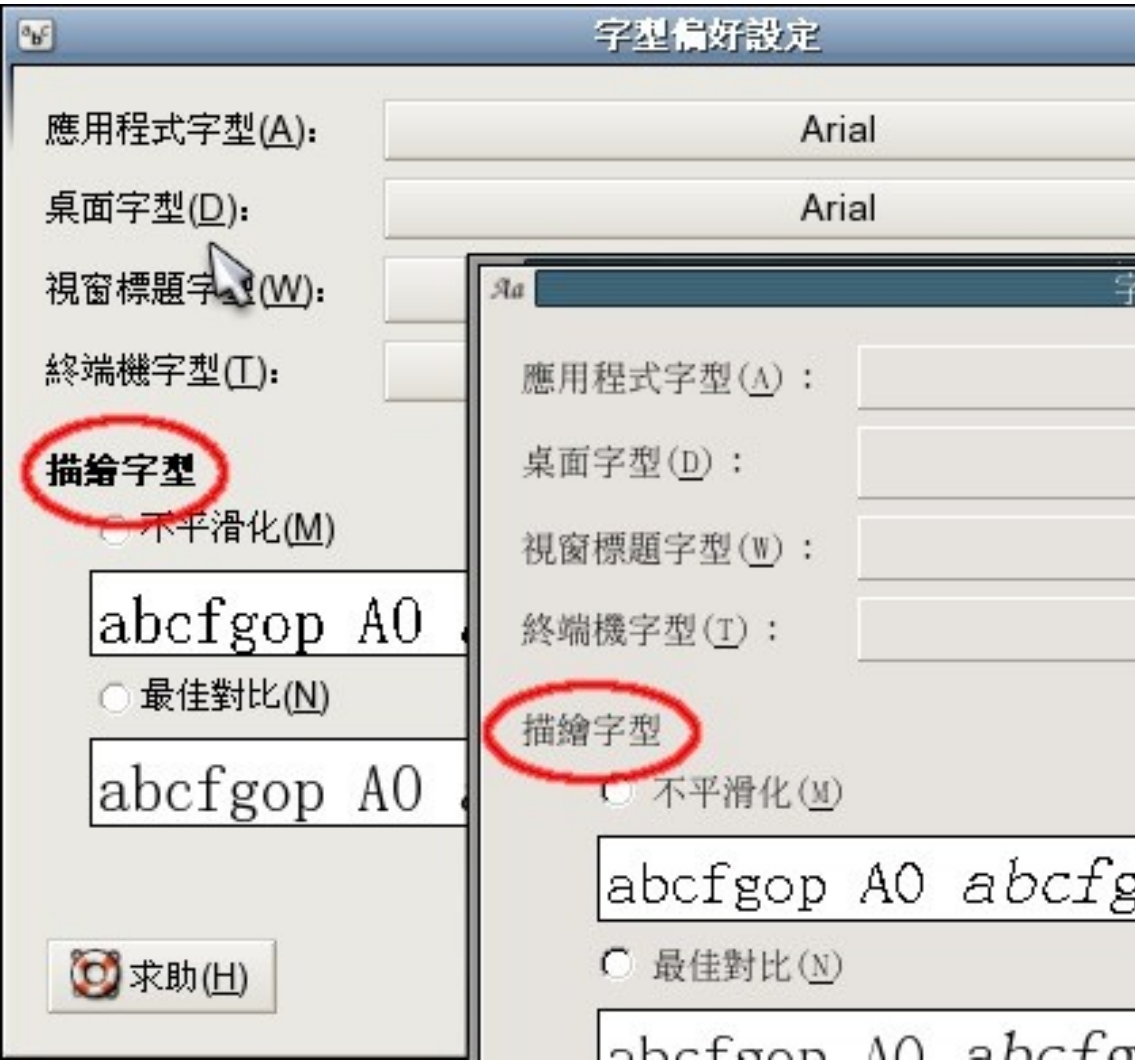
character 與 glyph

- Character (字元) != Glyph (字形)
- 隨著以下因素， character 或許會有不同的風貌：
  - regions (地區，如中港台日韓，漢字寫法不一)
  - character 在行文所處的位置（前、中、後）會影響最終的呈現形狀，如阿拉伯文
  - 與前後 character 的連接方式，會結合成新的形狀，如藏文或泰文

在 X 上的字型描繪處理

- 字型描繪方式
  - X core fonts
  - FreeType： 如 WINE
  - 應用程式自行實做，如 OpenOffice
- X11 core fonts 機制過於老舊
  - 只支援兩色 (黑 / 白) 字型顯示
  - 無法實做灰階 (greyscale) 或 Anti-aliasing
- 新途徑： Xft, Xft2/fontconfig, STSF 等

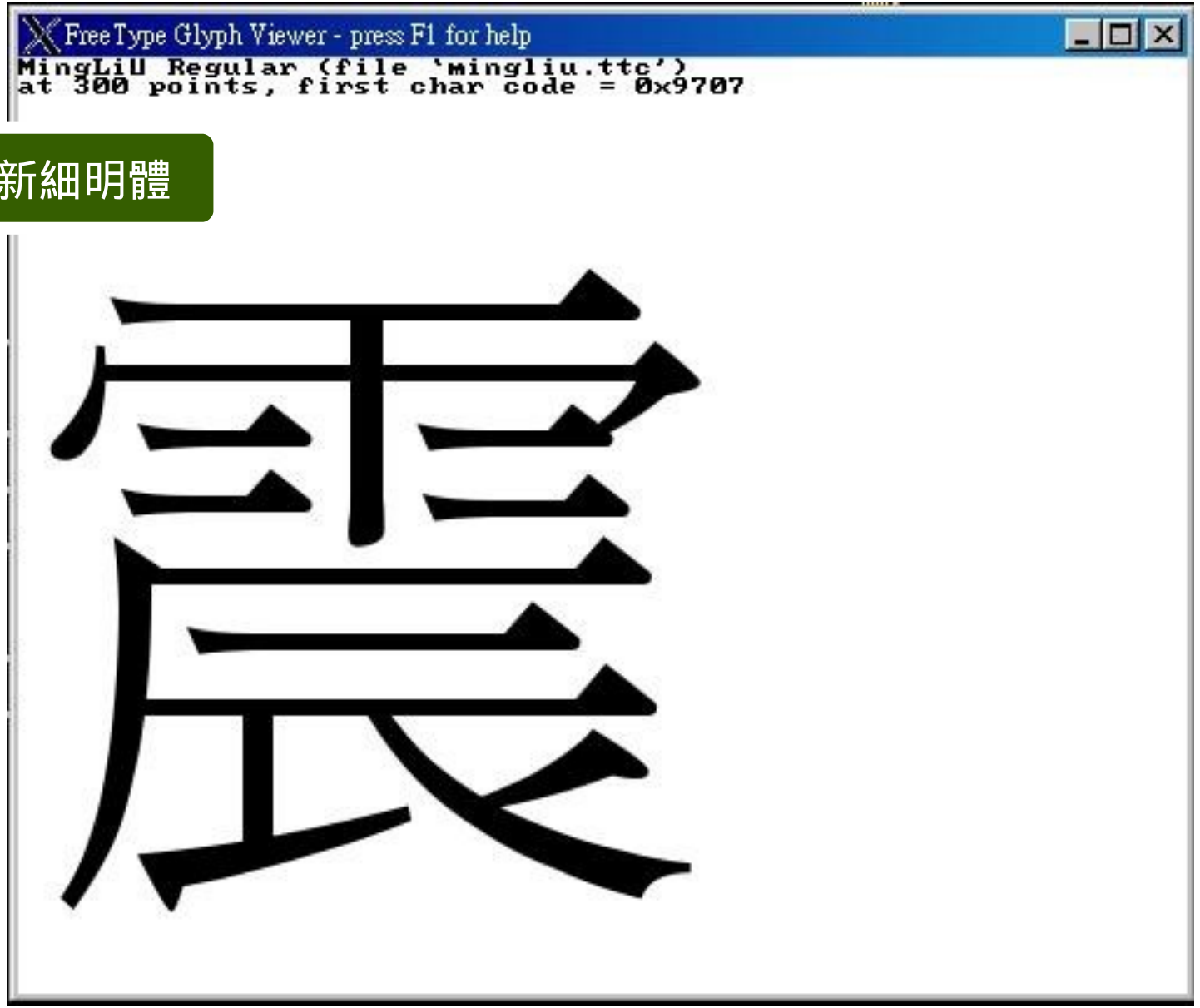
Xft/Fontconfig + FreeType



- FireFly 的貢獻
  - 最佳化中文顯示
  - 粗體 (Bold)
  - 斜體 (Itaic)
  - 中文字型名稱
- olv 的貢獻
  - optimizations to TrueType loader
  - glyph embolden support
  - CJK auto-hinter

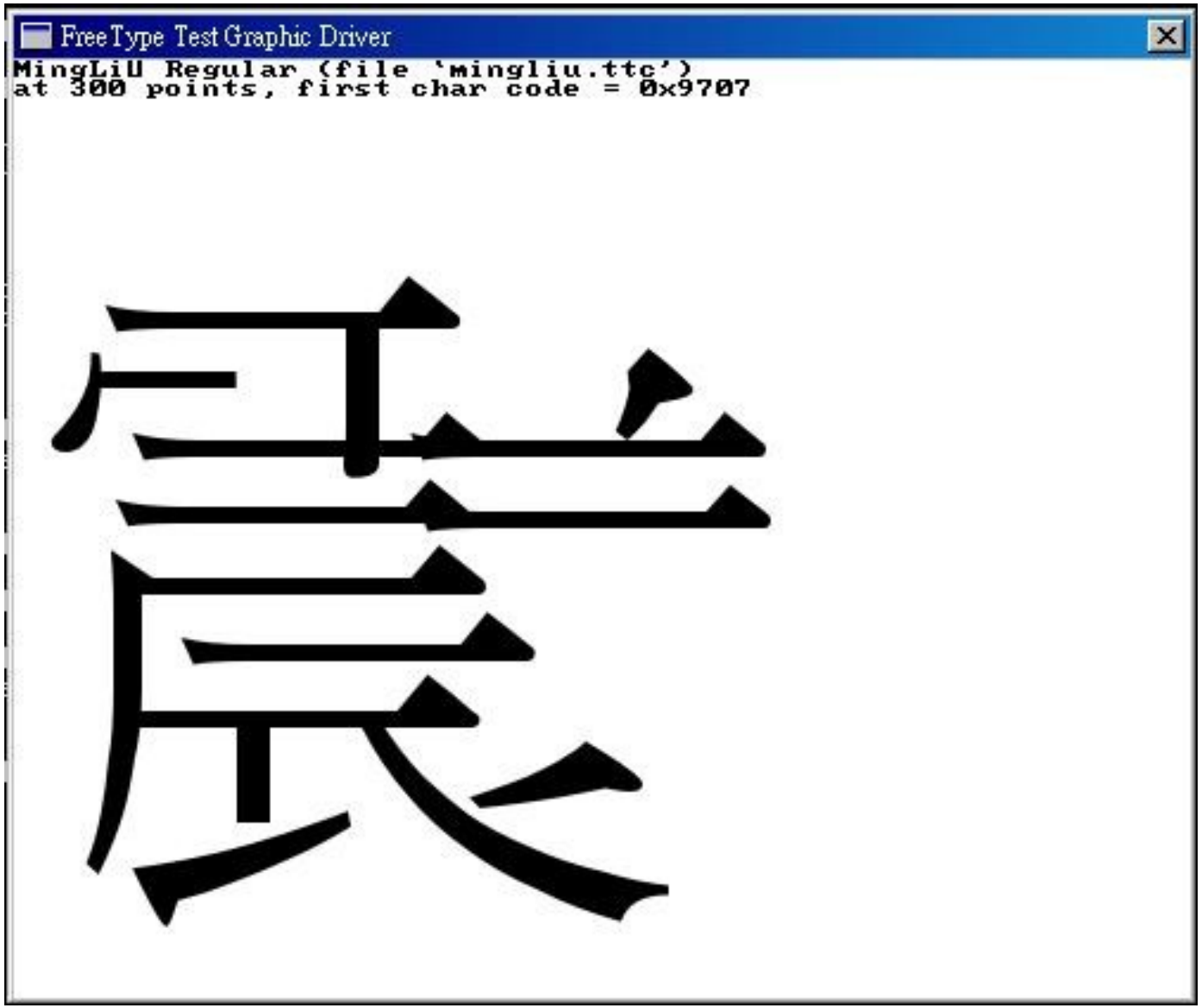
Normal Glyph

MingLiu = 新細明體



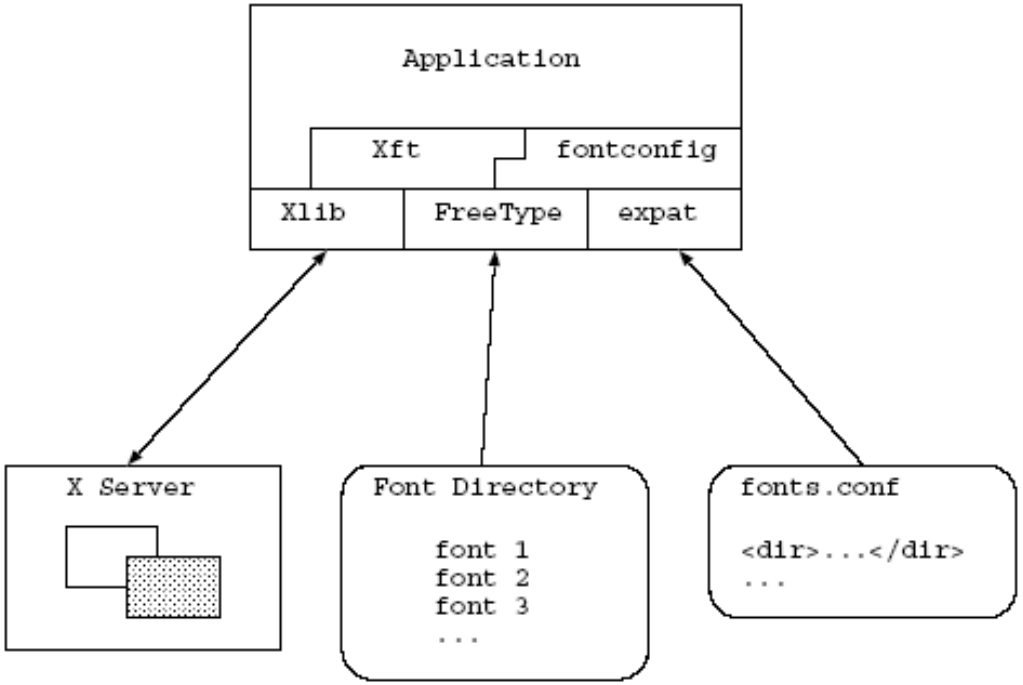
Broken Glyph

開啟 Bytecode Interpreter(針對筆畫組字字型)
(否則會破碎)



Xft/fontconfig

Figure 1. Fontconfig Application Architecture



- 透過 Render Extension
  - Alpha blending
  - anti-aliasing
  - sub-pixel
  - **server side**
- Xft Library
  - 以 FreeType 與 XRender library 作描繪動作
  - client side
- Fontconfig Library
  - Font accessing (**client side**)

Printing

- UNIX 的設計哲學影響到 Printing
  - 螢幕輸出與 Printing 是分離的
  - 預設使用 PostScript printers
- 缺乏 PostScript printer 的應用可使用 Ghostscript (GS)，但不支援從外界讀取 CJK 字型
- GS-CJK 成為 Ghostscript 的一部分，提供以下支援
  - PostScript CID / TrueTye CJK fonts
- Bg5ps 與 ttfprint 可解決部分問題

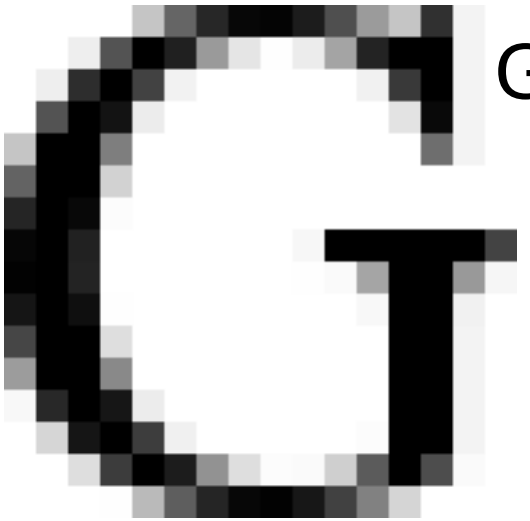
Quick fix / 非通解 → 全面移往 Unicode

GTK+ 多樣的 Rendering 機制

GDK

↓

Core X



GDK

↓

Xft

GDK → Pango

GDK

↓

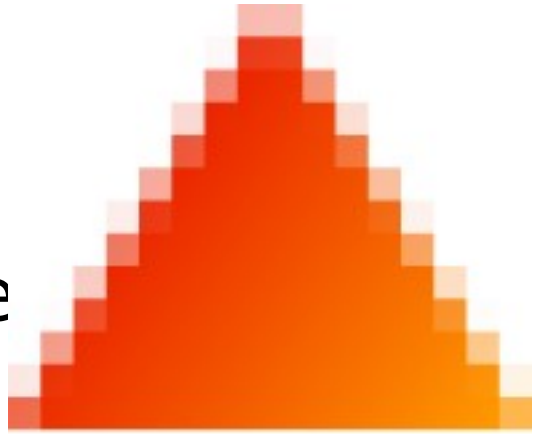
RENDER

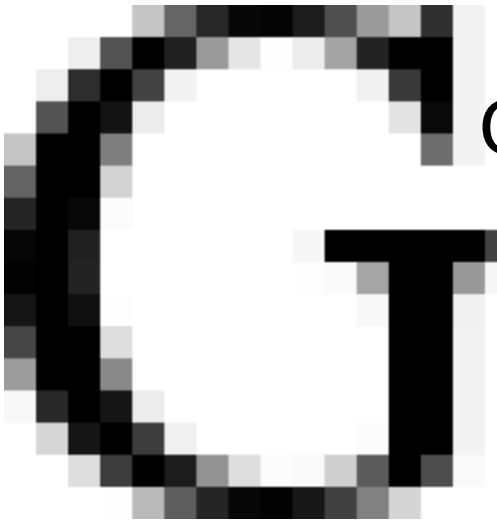
GTK+ 2.8 開發目標

GDK

↓

Mystery Gue





GDK

↓

Pango

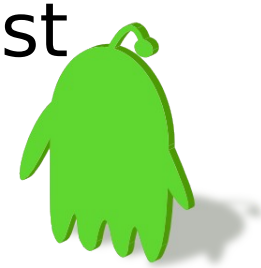
↓

Mystery Guest

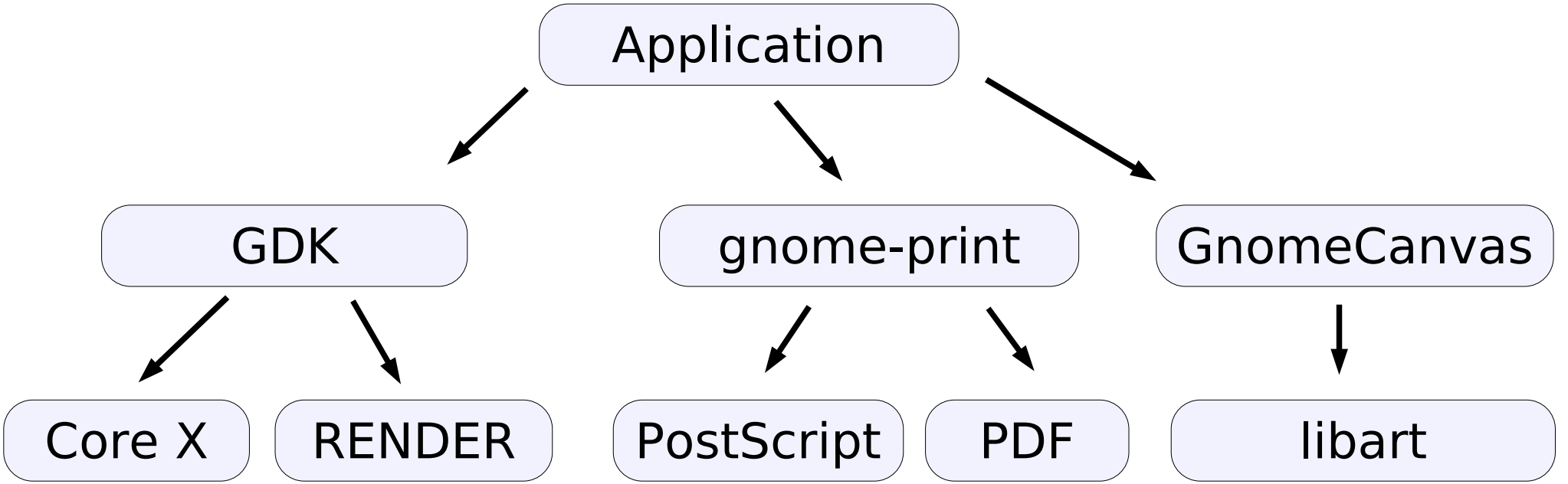
GDK

↓

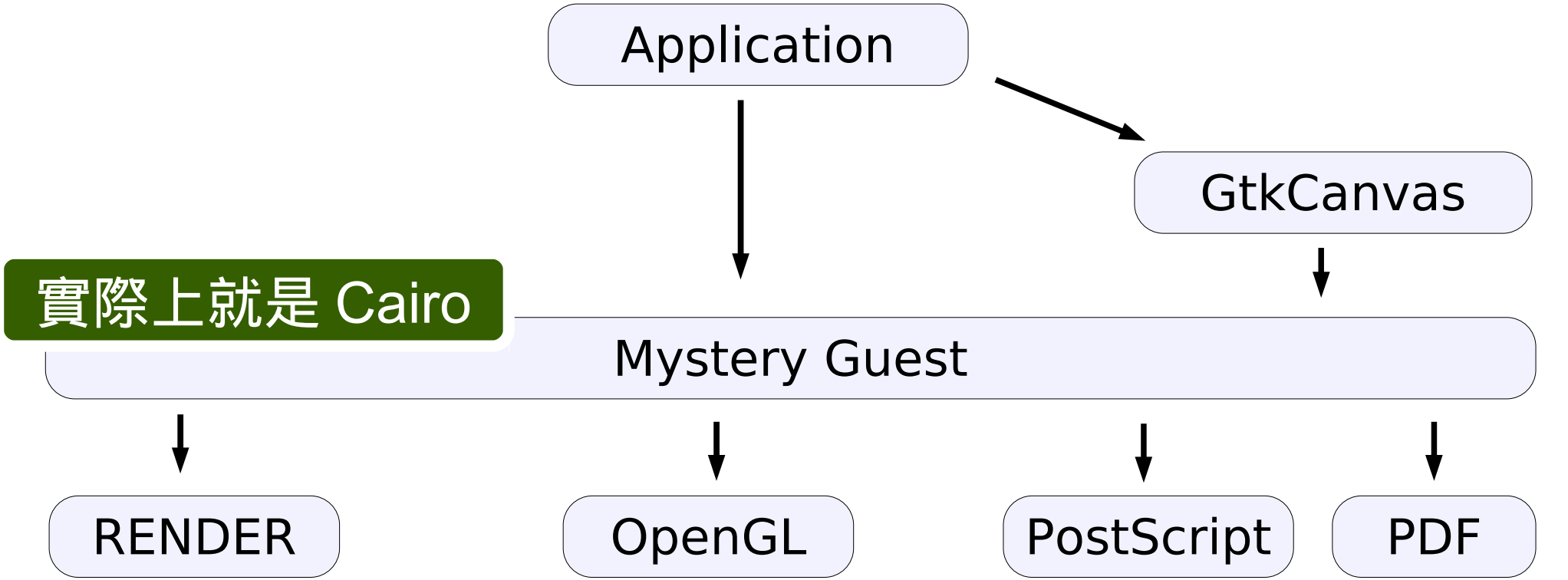
Mystery Guest



GTK+/GNOME 多樣的輸出介面



GTK+ 2.10 發展成果


Application
GtkCanvas
實際上就是 Cairo
Mystery Guest
RENDER
OpenGL
PostScript
PDF

自由的高品質中文字型

- 1999 年：文鼎科技 (Arphic) 釋出四套自由字型
- Arne Götje(高盛華) 合併 CJKUnifont
- 2004 年 10 月：Firefly 做出「新宋體」
- 2003 年：《香港增補字符集 - 2001》參考字型
- OAKA 開源香港常用中文字體計劃
- 研發天鸞字庫的台灣中原大學數學系王漢宗教授分別在 2000 和 2004 年捐出十套 WCL 系列字型和 32 套新字型 → 因授權議題處理不善，撤銷
- 文泉驛 (wqy) 致力創作簡體中文點陣字型
- ...to be continued...

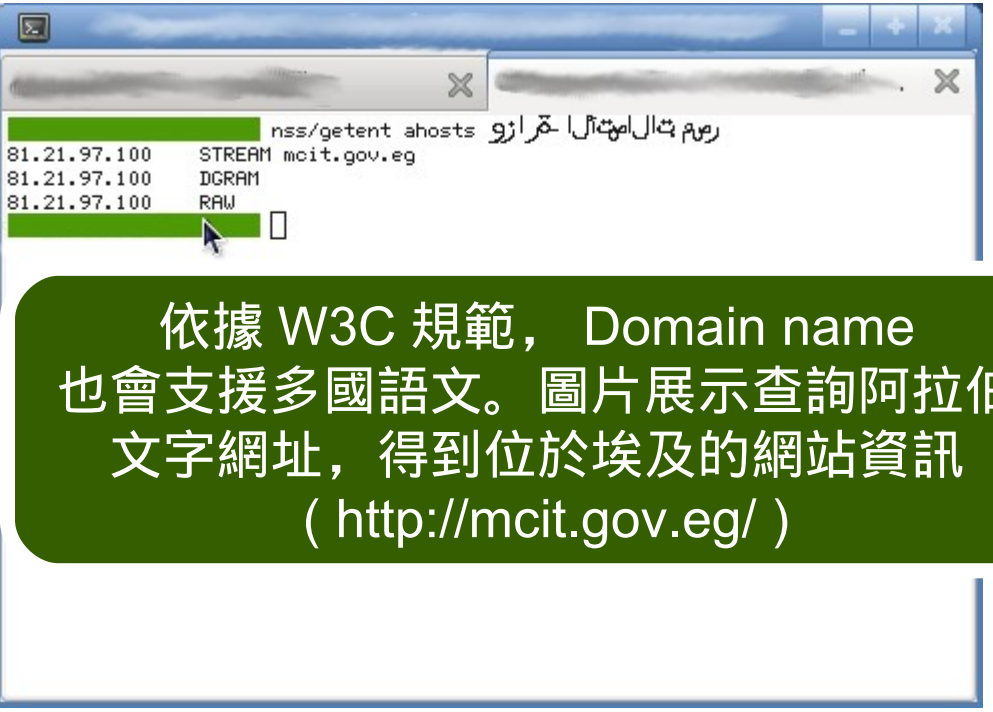
GNU 的基礎多國語文改良

(glibc 層面)

- 在 C 函式庫層面支援 multibyte 與 wide characters
- localedef 工具允許定義任何 charset 的區域性資訊，包含 UTF-8, UCS2
- 在 glibc 中， charset/encoding 的轉換表格以 shared object (DLL) 的形式存在
  - iconv() 函式
- wide character: UCS-4, native byte order

GNU 的基礎多國語文改良

(glibc 層面)



依據 W3C 規範， Domain name
也會支援多國語文。圖片展示查詢阿拉伯
文字網址，得到位於埃及的網站資訊
(http://mcit.gov.eg/)

- GNU libIDN
  - libidn 的目標是對 internationalized domain names 作編碼 / 解碼的處理
- GNU glibc 2.12 支援 IDN

除 GNU libc 外，亦有 IBM 主導的 ICU

- ICU = International Components for Unicode
- 開放原始碼，較寬鬆的授權模式，為其他專案採用，如 WebKit 與 Android
- 特徵
  - Text: Unicode 文字處理，完整的字元 / 字元集轉換 (500+ code pages)
  - Analysis: Unicode regular expressions; full Unicode sets; character, word and line boundaries
  - Comparison: 語言相關的比較與搜尋

區域性自由軟體 ／ 國際化發展模式

區域性自由軟體 (1)

- 輸入法系統 (非國際化架構)
  - gcin: 符合台灣地區使用者習慣的輸入法系統，主體以 Gtk+ 撰寫，支援 Gtk+/Qt im-module, XIM, Windows IME 等環境
  - oxim: 延續 XCIN 發展的輸入法系統，加入若干新的輸入法引擎與手寫辨識功能

- Unicode 終端機環境
  - UCIMF: 完整的 Unicode 輸出與輸入支援
- 輸入法引擎
  - libtabe: 詞音輸入法的核心

    http://libtabe.openfoundry.org
  - libchewing: 新酷音輸入法的引擎

    http://chewing.csie.net/
  - libfreearray: 自由行列輸入法引擎

    http://code.google.com/p/freearray/wiki/IbusFaft

「自由行列」含入 libchewing 的斷詞模組與符號輸入功能

區域性自由軟體 (2)

- 非典型圖形處理環境
  - SDL-im: 透過修改 libSDL 的方式，讓 SDL 的程式得以支援平台的輸入法 (X11, Windows IME)
- 輸入法框架
  - OpenVanilla: 跨越 MacOS X, GNU/Linux, Windows 等平台的輸入法框架，本體是「 Filter 」，亦即對輸出輸入作特定的狀態機處理
- 輸入法引擎
  - SunPinYin: 緣起 OpenSolairs 的子專案，現已獨立為支援多種平台的智能拼音系統
- 中文終端機
  - PCManX: 跨平台的 BBS 軟體，支援多個站台配置與獨特的功能

未來的機會

中文資訊處理 :: 未來的機會

- 現在的 GNU/Linux 環境有著堪用的機制
- 大環境的兩大趨勢
  - 移動運算
  - 雲端運算
- 就輸入法來說，可建立專屬的雲端服務，讓使用者更新遠端的自訂詞庫 (新詞 / 修訂錯誤 / 流行詞彙 / 詞頻 / 詞性)，進而改善輸入法的品質。而且這種服務可讓倉頡、行列、大易等輸入皆受益
- 手寫輸入的辨識不該拘泥於完整字的處理，事實上，注音符號的連寫搭配智慧型輸入法引擎，如新酷音，就可用相對少的運算資源 (僅需辨識出 1409 種組合)，做出充分輸入動作，還可配合前述的雲端服務

http://0xlab.org