Chinese World

hmwang

編碼標準 Encoding Standard

- 電腦是美國人發明的
  - ASCII (American Standard Code for Information Interchange)
- 地方的電腦也要顯示中文
  - Big5
  - 台灣財團法人資訊工業策進會 在 1983 年為 五大中文套裝軟體 設計的編碼系統
  - 繁體中文中最常用的電腦中文字符集標準
  - 萬年遺毒

編碼標準

❑ ASCII

- 8 bits (理論上有 256 種可能)
- 0x00 ~ 0x7F 共 128 種字元
  - 0x00 ~ 0x1F ➔ control characters
  - 0x20 ~ 0x7F ➔ printable characters

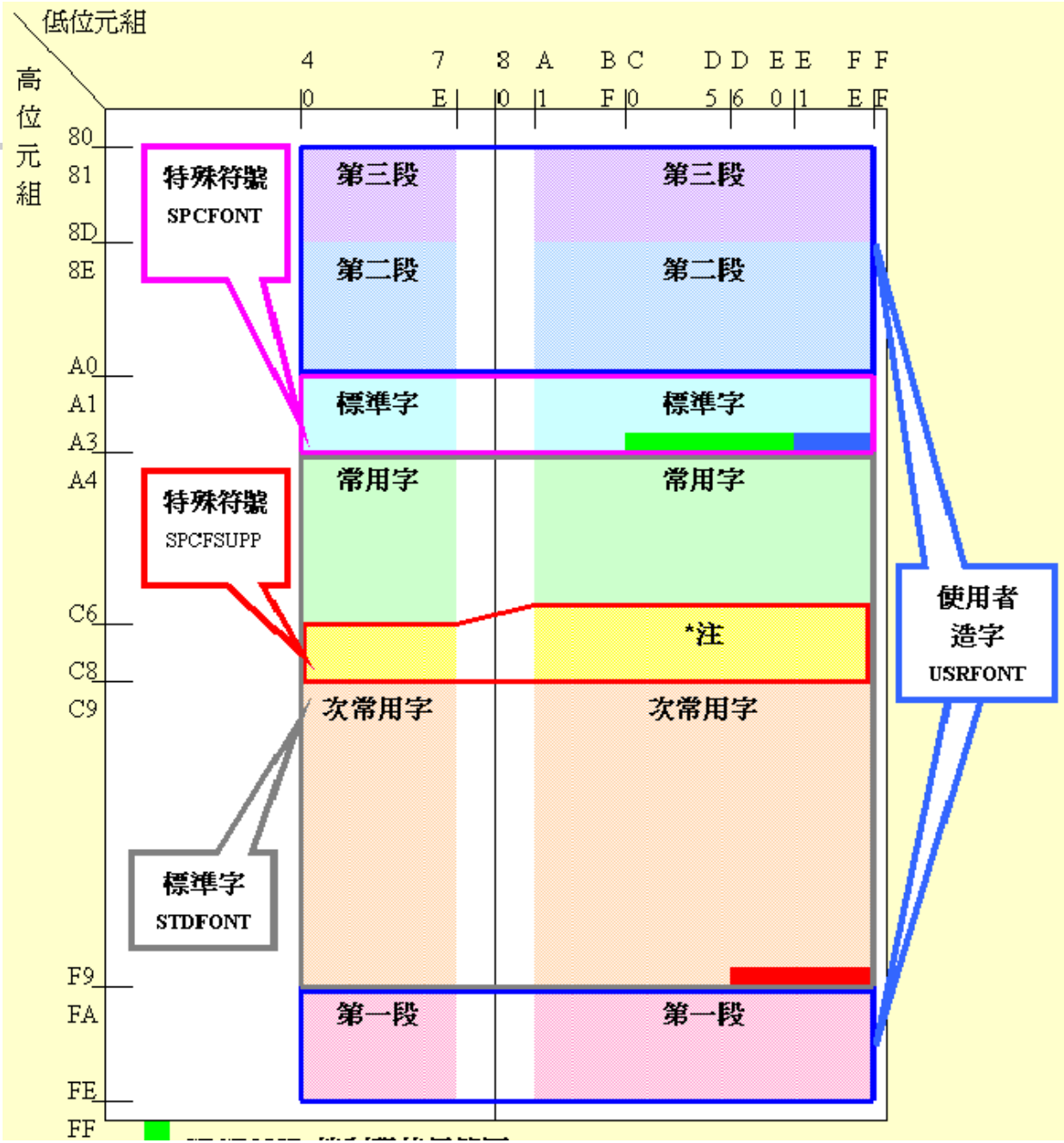
❑ Big5

- 使用 2 bytes 來存放中文字 (理論上有 65536 種可能)
- 實際上為與 ASCII 相容, 只能使用 19782 個
  - [0x81 ~ 0xFE][0x40~0x7E, 0xA1 ~ 0xFE]
    = 126 * (63 + 94) = 126 * 157 = 19782

Ref: http://www.cns11643.gov.tw/AIDB/encodings.do

編碼標準 – Big5

- 標準字 (13502)
  - 常用字
    - 你我他的媽
  - 次常用字
    - 杓晃束鐰飉
- 特殊符號 (441)
  - 符號、控制碼
    - ：！。∩ ⇳ ⊦
  - 罕用符號
- 使用者造字 (5809
  - 三段



Big5的問題

- 使用者造字區
  - 每個人都可以自己造字
    於是自己造的字放到別人電腦上就看不到
- 缺字
  - 塋、煊、栢、喆
- 延伸版本繁雜
  - 倚天Big5延伸
  - Code Page 950
  - Big5+
  - 族繁不及備載..
- 許功蓋問題 (\)
  - 0x5C (\) 會有特殊意義
  - 許（0xB35C）、功（0xA55C）、蓋（0xBB5C）

編碼標準 - Unicode

全世界共有上百種文字，因此有很多種不同的編碼系統

日本有 JIS，中國有 GB 2312，... etc

同樣的編碼在不同的編碼系統下顯示會不同

Unicode 組織就誕生了!!!!

Unicode VS ISO 10646

- 1991年左右，同時有兩個組織著手規範世界字碼
  - Unicode
  - ISO 10646

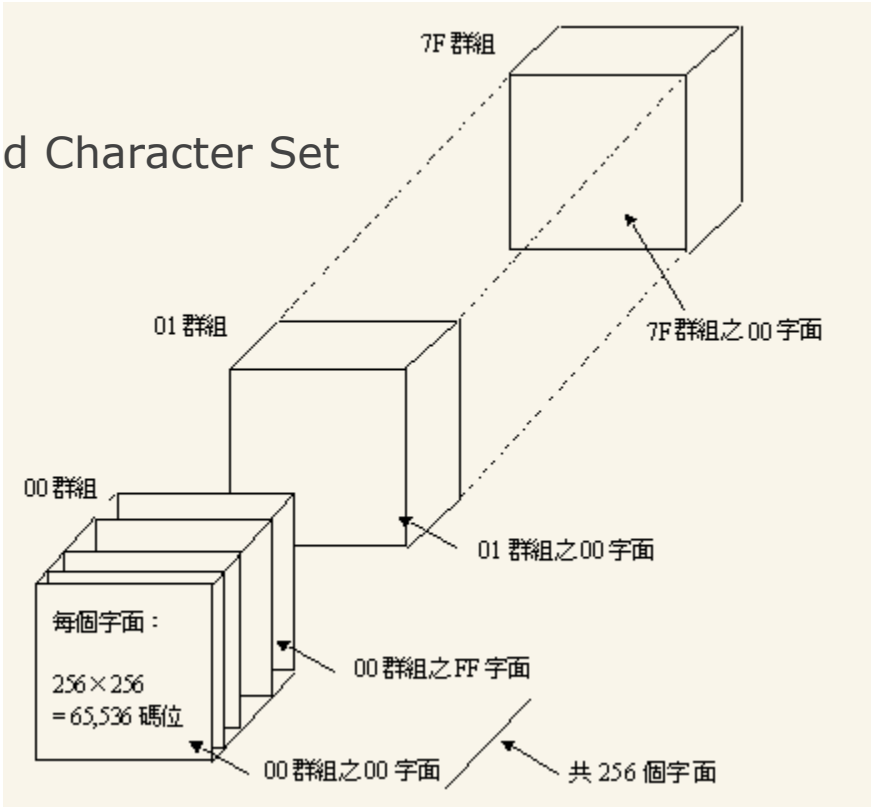
- 過不了多久，他們就互相體認到
  「這個世界不需要兩套不同的單一字符集」
  - 因此他們決定共用同樣的字碼

- 現在這兩個組織各自存在，各自互相砥礪

ref:http://zh.wikipedia.org/wiki/ISO_10646

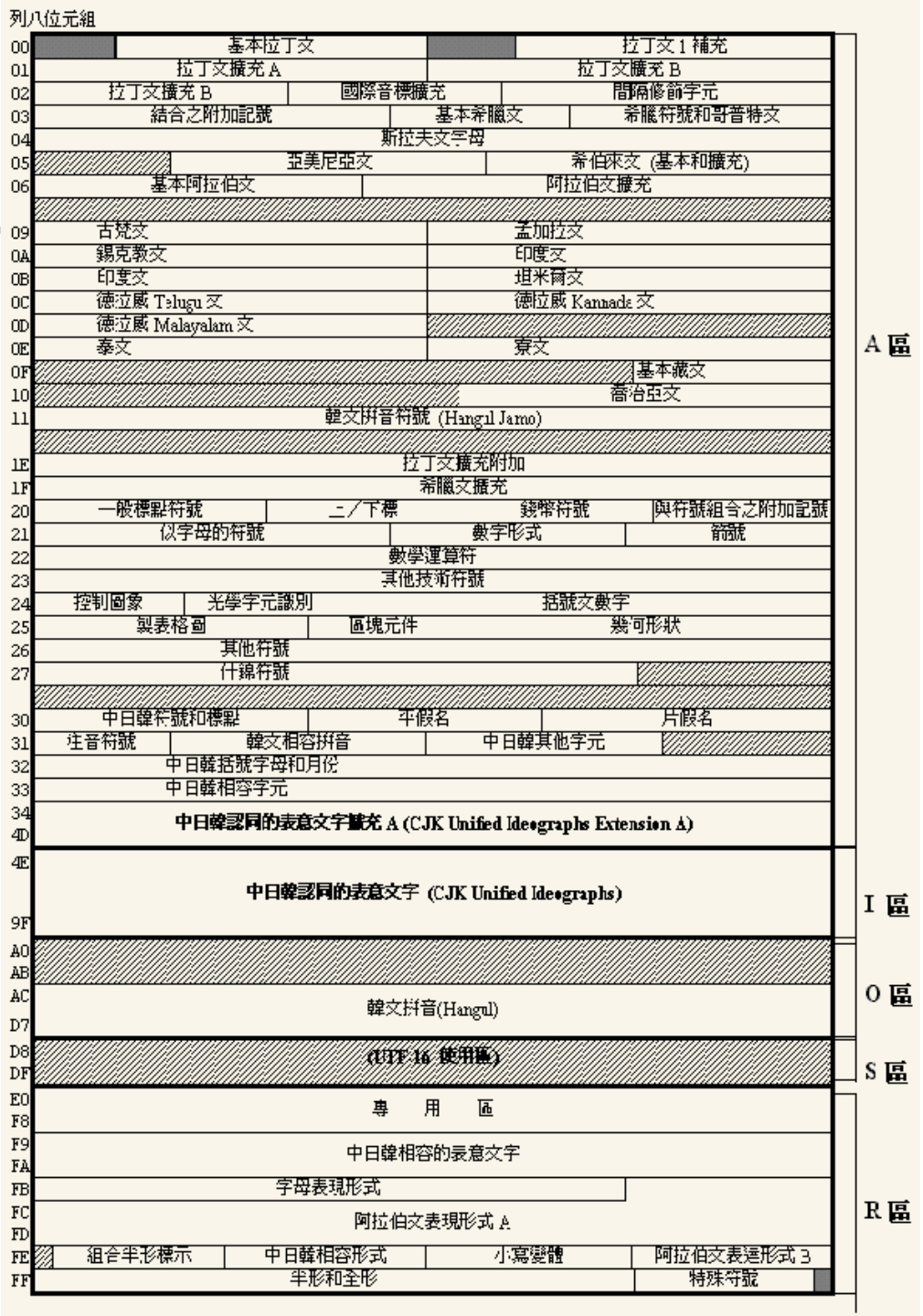
編碼標準 – ISO10646 and Unicode (1)

- Goal
  - 集結全球通用字符集,成一大聯集
  - UCS-4
    - Universal multiple-octet coded Character Set
  - 4 bytes encoding (2^31)
    - 128 Groups
    - 256 Planes each group
    - 256 Rows each plane
    - 256 Cells each row
  - BMP (UCS-2)
    - Basic Multilingual Plane
    - 00 group, 00 plane
    - 65536 encoding space
  - Why in BMP
    - 若所有字集都在 **BMP** 中, 就可以只使用 **2 bytes,** 否則就要用 **4 bytes,** 不能混用



編碼標準 – ISO10646 and Unicode (2)

- BMP



Unicode 的問題

- Big Endian & Little Endian
  - U+4E59？（乙）
  - U+594E？（奎）
- 編碼空間浪費
  - ASCII 字元通通都用 2byte 表示：0x00 0x41 「A」第一位永遠是0

Unicode Transformation Format

- UTF: UCS/Unicode Transformation Format
  - UTF-16(2、4 bytes)
    - 將一個 32-bit ISO10646 字元轉成多個 16-bit Unicode
    - Windows
  - UTF-8(1~4 bytes)
    - 將一個32-bit ISO10646 字元轉成多個 8-bit Unicode
    - 將一個16-bit Unicode 字元轉成多個 8-bit Unicode
    - 128個US-ASCII字元只需1 bytes編碼
    - 帶有附加符號的拉丁文、希臘文、西里爾字母、亞美尼亞語、希伯來文、阿拉伯文、敘利亞文及它拿字母則需要 2 bytes 編碼
    - 其他基本多文種平面（BMP）中的字元（這包含了大部分常用字）使用 3 bytes 編碼
    - 其他極少使用的 Unicode 輔助平面的字元使用 4 bytes 編碼
    - Unix-like systems

非常經典的 UTF-8...

- 與既有系統的相容性
  - 只包含 ASCII 0-127 的字串是合法的 UTF-8 字串
  - NULL-terminated 字串處理
- 極高的辨識性
  - UTF-8字串可以由一個簡單的演算法可靠地識別出來。
- 可以容納所有 Unicode 字元
  - UTF-8 理論值可以容納百萬個字元 (實際是 1112064 個)
  - (2012 年發佈的 Unicode 6.2 也才十一萬個字元)
- Unicode 與 UTF-8 之間的轉換很方便

中文環境 (1)

- 要做到哪些事情
  - 中文訊息
  - 中文顯示
  - 中文輸入
  - 中文列印
  - 中文處理

簡單

困難

中文環境 (2)

- 中文化方式
  - 直接修改程式
    - 套件以排山倒海之勢而來只有真強者才能改完了Orz

      18 chars
  - 國際化(InternationalizatioN，i18n)
    - Multi-language architecture
      - 程式設計人員按照該架構的機制與準則寫程式, 便可支援各式各樣的語言
    - Locale (LOCALization Environment database)
      - 程式根據使用者選擇的 locale 聯繫到不同資料庫, 進而提供該語言的支援
  - 在地化(LocalizatioN，L10n)
    - 在 i18n 的大架構下，加入「在地化」的特性
  - 通常i18n只需做一次，而L10n要針對每個語言個別做

i18n & L10n

- i18n + L10n
  - 語言/翻譯
  - 文化、書寫習慣
    - ➢ 名字和稱謂的位置
    - ➢ 電話號碼，位址和國際郵遞區號的格式
    - ➢ 貨幣單位
    - ➢ 度量衡
    - ➢ 日期時間
    - ➢ 時區
    - ➢ 數字格式
- L10n only
  - 內容在地化
  - 道德在地化
  - 文化價值
  - 社會環境

中文環境 (3)

- locale in FreeBSD
  - 地區性語言的資訊
    - LC_ALL
    - LC_CTYPE
      - 掌管該 locale 中所有字元的處理方式
    - LC_MESSAGES
      - 掌管程式訊息輸出所用的語言
    - LC_TIME
      - 時間格式
    - LC_NUMERIC
      - 數字格式
    - LC_MONETARY
      - 貨幣格式
    - LC_COLLATE
      - 字母順序與特殊字元比較
    - LANG
      - 語言顯示
  - 效力優先性：LC_ALL > LC_* > LANG

中文環境 (4)

- 設定 locale
  - csh/tcsh shell
    - setenv LC_CTYPE en_US.UTF-8
  - Bourne Shell
    - export LC_CTYPE=en_US.UTF-8

Note: 可以寫在 .tcshrc/.bashrc 中登入後自動載入

  - /usr/share/locale/
    - 各國的 locale 資訊
    - 命名規則: 語言_地區名.字元編碼名稱
      - zh_TW.UTF-8
      - zh_CN.GBK

中文環境 (5)

- 中文 Terminal (Remote Login)
  - M$ Windows: putty, pietty, netterm, multi-term, telnet, …etc.
  - X Window: xterm, rxvt, aterm, mterm,roxterm…etc.
  - 設定好中文支援，登入後即可看到中文
    - setenv LC_CTYPE en_US.UTF-8 (csh/tcsh)
    - export LC_CTYPE=en_US.UTF-8 (sh/bash)
    - 顯示為英文但支援 multibyte characters
- 中文 Xwindow
  - 建立支援 L10n 中文環境
    - 安裝中文字型
    - 設定 Shell locale 環境
    - 安裝中文輸入法 (Ex. ibus)

Steps

- 安裝中文字型
- 安裝中文 Terminal Emulator
- 安裝中文輸入法 (Ex. ibus)
- 其他設定

安裝中文字型 (1)

- 兩大中文字型種類
  - 點陣字型 (Bitmapped Font)
    - BDF (Bitmap Distribution Format) 點陣分散格式
    - HBF (Hanzi Bitmap Font) 漢字點陣字體
    - PCF (Portable Compiled Font)
  - 曲線描邊字型 (Outline Fonts)
    - True Type Font (TTF)

安裝中文字型 (2)

- Font Path
  - % xset q
  - % xset fp+ [directory]
  - % xset fp rehash

Font Path:
/usr/local/lib/X11/fonts/misc/,
/usr/local/lib/X11/fonts/TTF/,
/usr/local/lib/X11/fonts/Type1/,
/usr/local/lib/X11/fonts/75dpi/,
/usr/local/lib/X11/fonts/100dpi/,
/usr/local/lib/X11/fonts/local/

- 安裝字型
  1. 直接從 Windows 下偷過去
  2. 透過 ports 安裝字型檔案
  - 使用 fc-cache 建立字型資料庫
  - 修改各軟體設定使用別的字型

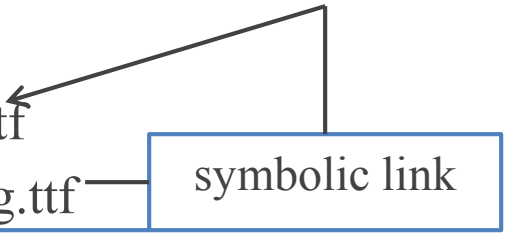
安裝中文字型 (3)

Fireflyttf

- 透過 ports 安裝的都會自己跑
  - % ttfm.sh --add xttfm /usr/local/share/fonts/TrueType/fireflysung.ttf
  - % fc-cache –f –v /usr/local/lib/X11/fonts/TrueType/
- portmaster chinese/fireflyttf

安裝中文字型 (4)

- 增加 Font Path
  - Edit /etc/X11/xorg.conf
    - /usr/local/share/fonts/TrueType/fireflysung.ttf
    - /usr/local/lib/X11/fonts/TrueType/fireflysung.ttf



  - Restart xwindow

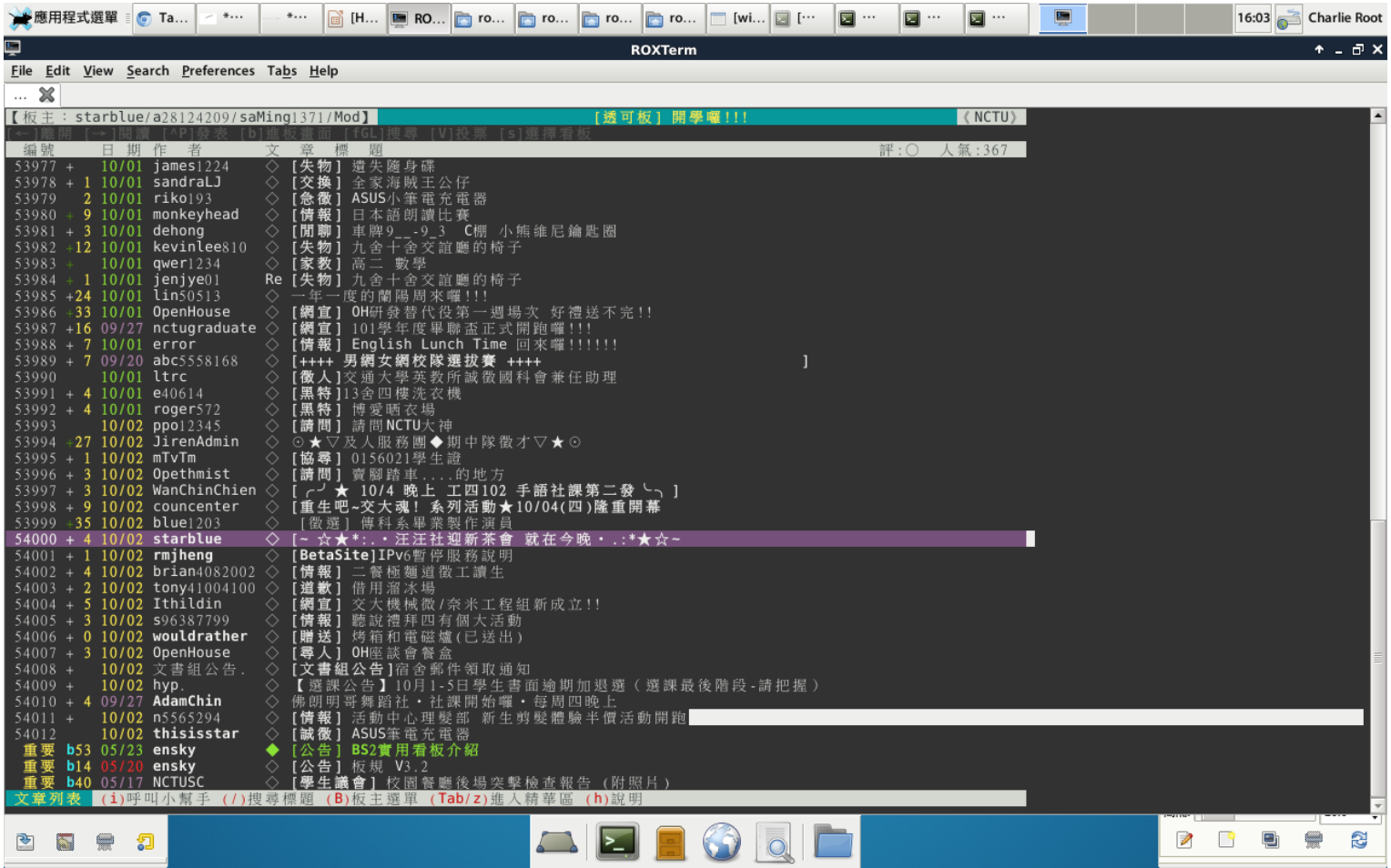
```
Section "Files"
        ModulePath   "/usr/local/lib/xorg/modules"
        FontPath     "/usr/local/lib/X11/fonts/misc/"
        FontPath     "/usr/local/lib/X11/fonts/TTF/"
        FontPath     "/usr/local/lib/X11/fonts/OTF"
        FontPath     "/usr/local/lib/X11/fonts/Type1/"
        FontPath     "/usr/local/lib/X11/fonts/100dpi/"
        FontPath     "/usr/local/lib/X11/fonts/75dpi/"
        FontPath     "/usr/local/lib/X11/fonts/TrueType/"
        FontPath     "/usr/local/lib/X11/fonts/local/"
EndSection
```

安裝中文 Terminal

- rxvt-unicode
  - /usr/ports/x11/rxvt-unicode
- aterm
  - /usr/ports/chinese/aterm
- eterm
  - /usr/ports/chinese/eterm
- ROXterm
  - /usr/ports/x11/roxterm
- mlterm
  - /usr/ports/x11/mlterm

ROXterm

- X11/rxvt-unicode
- roxterm-config



安裝中文輸入程式

- Choices
  - ibus-chewing(chinese/ibus-chewing)
  - ibus-pinyin(chinese/ibus-pinyin)

安裝 ibus 中文輸入程式 (1)

- ibus
  - Intelligent Input Bus

1. % cd /usr/ports/textproc/ibus-chewing ; make install clean
2. setenv LC_CTYPE zh_TW.UTF-8 (csh/tcsh)
   export LC_CTYPE=zh_TW.UTF-8 (sh/bash)
3. Edit .xinitrc(或是可以 setenv in .cshrc/.bashrc)

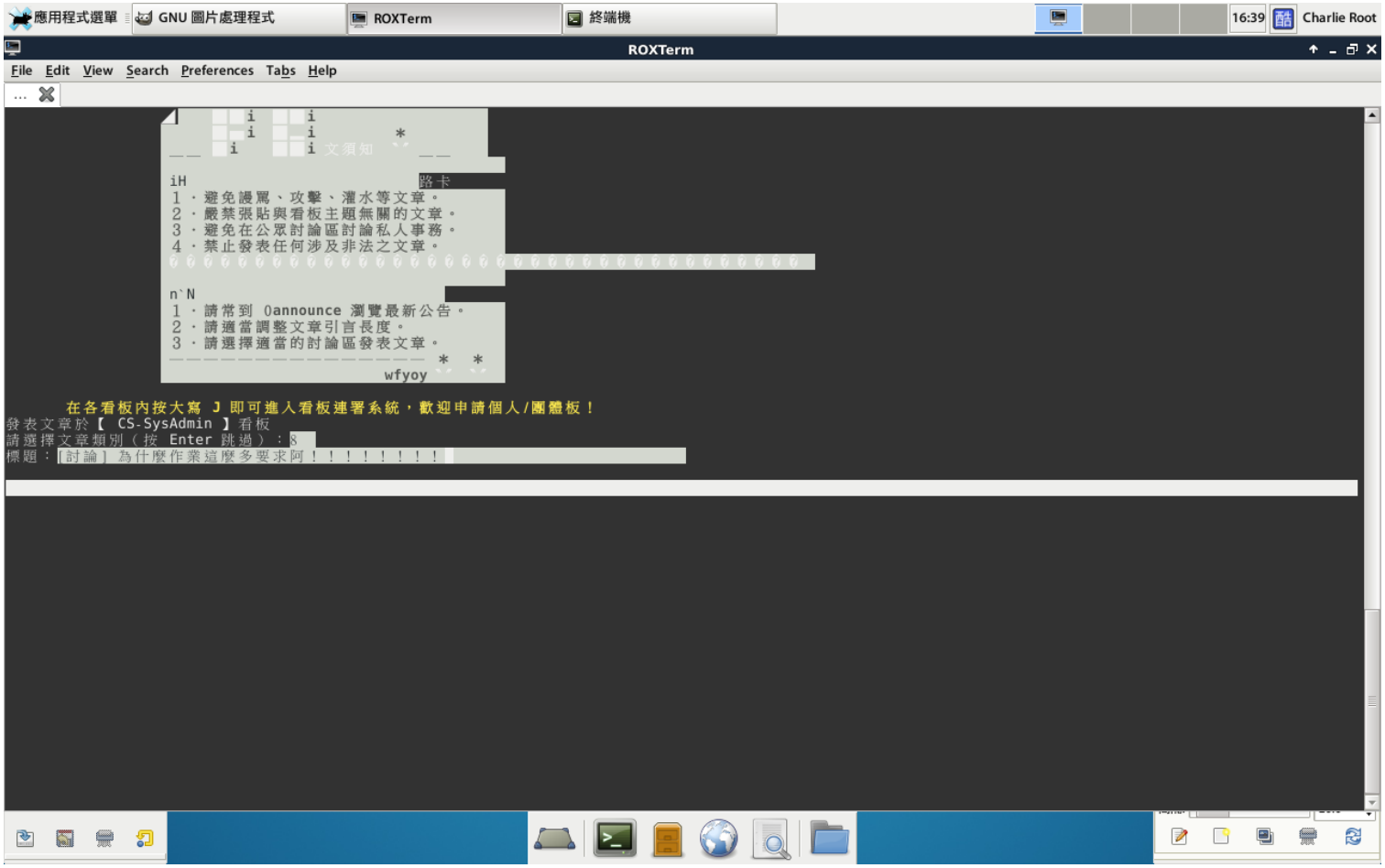
```
XIM=ibus
GTK_IM_MODULE=ibus
QT_IM_MODULE=xim
XMODIFIERS='@im=ibus'
XIM_PROGRAM="ibus-daemon"
XIM_ARGS="--daemonize --xim"
```

安裝 ibus 中文輸入程式 (2)

- ibus相關設定
  - % ibus-setup (UTF-8)
  - 可以加入 Chewing

安裝 ibus中文輸入程式 (3)

4. Switch to chinese input: Ctrl -Space



References

- 中文碼介紹
  - http://www.cns11643.gov.tw/web/word.jsp
- FreeBSD Chinese HOWTO
  - http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/index.html
- Introduction to i18n
  - http://www.debian.org/doc/manuals/intro-i18n/
- Unicode 介紹
  - http://www.csie.ntu.edu.tw/~p92005/Joel/Unicode.html