

第五章

使用者中文輔助系統之建置

王乃誼

國立中央大學資訊管理學系三年級
im801085@mgt.ncu.edu.tw

朱勇政

國立中央大學資訊管理學系二年級
im811095@mgt.ncu.edu.tw

林秉忠

國立中央大學資訊管理學系二年級
im811100@mgt.ncu.edu.tw

5、1 節 XFree86之建置

5.1.1 前言：

視窗系統(Window System)最早源於一九七〇年Xerox PARC開發的Xerox star資訊系統。X視窗係由美國麻省理工學院(M.I.T)與DEC、HP、與IBM等數家大電腦公司所開發的一套視窗軟體。它以分散式的"委託者／伺服器"(Client/Server)模式為架構，具備網路透通性(Network Transparency)與設備無關性(device independent)等特性。

5.1.2 簡介：

XFree86是依據X視窗系統X11R5(X Window System, Version 11, Release 5)在以Intel公司的CPU為基底(Intel-based)機器上不同版本UNIX作業系統的移植版本。此次安裝的版本為XFree86 2.0。XFree86 2.0 在能力上比先前的Release向前跨了一大步。有大量的新程式碼，包括全新的伺服器，以及許多其他方面的增強。

5.1.3 安裝步驟：

5.1.3.1 抓取檔案

此次安裝的XFree86 2.0的可執行檔是從中央大學無記名(Anonymous)FTP站 softu1.src.ncu.edu.tw中的/pub4/XFree86/XFree86-2.0目錄取得。這些壓縮檔的內容包含：

基本要求：

- bin - 所有可執行的X委託者應用程式
- fonts - 雜項(misc)與75 dpi字型
- doc - README檔，安裝敘述，與XFree86指定的線上輔助頁(man page)
- lib - 執行時所需的資料檔
- inc - 圖像(icon)的二元圖(bitmap)等
- config - 依客戶情況的xinit，xdm，與fs執行時的構形(configuration)檔

配合自己的顯像硬體選擇以下之一：

- 8514 - IBM 8514/A卡伺服器
- Mach32 - ATI Mach32圖形加速卡伺服器
- Mach8 - ATI Mach8圖形加速卡伺服器
- Mono - VGA，Super-VGA，Hercules，與其他顯像卡伺服器
- S3 - 以S3為基底顯像卡伺服器
- SVGA - Super-VGA卡的8位元虛擬色彩(pseudocolor)伺服器
- VGA16 - VGA與Super-VGA卡的4位元虛擬色彩伺服器

擇性的：

- man - 除了XFree86指定之外的線上輔助頁
- fonts-100dpi - 100 dpi字型
- fonts-scaled - Speedo與Type1字型
- prog - 只在編譯(compile)時需要的lib*.a與*h檔
- xdm-des - 有des封裝(encryption)的xdm可執行檔
- xdm-nodes - 沒有des封裝的xdm可執行檔
- linkkit - X伺服器重構形工具(reconfiguration kit)
- fontserver - 字型伺服器與其線上輔助頁
- pex - pex委託者、函數庫(libraries)、標頭檔(header file)與資料

5.1.3.2 處理壓縮檔案

至少必須安裝bin、fonts、doc、lib、config與配合顯像卡的伺服器。樣本機器是將XFree86的壓縮檔置於/usr/augustin目錄下，然後使用下列方式解壓縮：

```
cat XFree86-2.0-bin.tar.gz.* | gunzip | tar xf -
cat XFree86-2.0-doc.tar.gz | gunzip | tar xf -
cat XFree86-2.0-fonts-100dpi.tar.gz | gunzip | tar xf -
cat XFree86-2.0-fonts-scaled.tar.gz | gunzip | tar xf -
cat XFree86-2.0-fonts.tar.gz | gunzip | tar xf -
cat XFree86-2.0-fontserver.tar.gz | gunzip | tar xf -
cat XFree86-2.0-lib.tar.gz | gunzip | tar xf -
cat XFree86-2.0-inc.tar.gz | gunzip | tar xf -
cat XFree86-2.0-linkkit.tar.gz | gunzip | tar xf -
cat XFree86-2.0-man.tar.gz | gunzip | tar xf -
cat XFree86-2.0-pex.tar.gz | gunzip | tar xf -
cat XFree86-2.0-prog.tar.gz | gunzip | tar xf -
cat XFree86-2.0-xdm-des.tar.gz | gunzip | tar xf -
cat XFree86-2.0-xdm-nodes.tar.gz | gunzip | tar xf -
cat XFree86-2.0-config.tar.gz | gunzip | tar xf -
cat XFree86-2.0-8514.tar.gz | gunzip | tar xf -
cat XFree86-2.0-Mach32.tar.gz | gunzip | tar xf -
cat XFree86-2.0-Mach8.tar.gz | gunzip | tar xf -
cat XFree86-2.0-Mono.tar.gz | gunzip | tar xf -
cat XFree86-2.0-S3.tar.gz | gunzip | tar xf -
cat XFree86-2.0-SVGA.tar.gz | gunzip | tar xf -
cat XFree86-2.0-VGA16.tar.gz | gunzip | tar xf -
```

解壓縮之後，會在現在目錄/usr/augustin下產生X386的子目錄。

5.1.3.3 安裝

執行安裝程式/usr/augustin/X386/lib/X11/etc/x386install，它會產生/usr/X386的連結(symbolic link)指向/usr/augustin/X386，並檢查/dev與/etc/ttys的內容，然後將/usr/X386/bin加入/etc/csh.login的預設搜尋路徑(default path)中。

依自己的顯像介面卡建立一名為X的連結檔。例如，樣本機器的顯像卡為ET4000，因此以下列方式建立：

```
cd /usr/X386/bin      - 進入該目錄
rm X                  - 刪除原來的連結
ln -s XF86_SVGA X    - 建立新連結(ET4000卡屬於Super-VGA)
```

此外，確定將/usr/X386/bin加入自己shell的path變數。

5.1.3.4 修改配備檔

依自己的硬體情況建立/usr/X386/lib/X11/Xconfig檔案。在該目錄下原有一份Xconfig.sample的範例檔，可直接複製成Xconfig再修改，或是命名為Xconfig.hostname(如Xconfig.athena2.net.ncu.edu.tw)成為本機器的專用設備檔。樣本機器修改後的內容如下：

```
# $XFree86: mit/server/ddx/x386/Xconfig.cpp,v 2.0 1993/10/08 15:55:00 dawes
#      Exp $
# $XConsortium: Xconfig,v 1.2 91/08/26 14:34:55 gildea Exp $
#
# Copyright 1990,91 by Thomas Roell, Dinkelscherben, Germany.
# Copyright 1992,93 by David Dawes, David Wexelblat
#
# Author: Thomas Roell, roell@informatik.tu-muenchen.de
#
# Extensive modifications by the XFree86 Core Team
#
*****
# Refer to the Xconfig(4/5) man page for details about the format of
# this file. This man page is installed as /usr/X386/man/man5/Xconfig.5x
#
*****
```

```

#
*****
# Generic parameters section 一般參數
#
*****
#
# some nice paths, to avoid conflicts with other X-servers
#
# RGB資料庫檔案的路徑
RGBPath    "/usr/X386/lib/X11/rgb"
#
# Multiple FontPath entries are allowed (which are concatenated together),
# as well as specifying multiple comma-separated entries in one FontPath
# command (or a combination of both methods)
#
# 預設字型的路徑
FontPath "/usr/X386/lib/X11/fonts/misc/"
FontPath "/usr/X386/lib/X11/fonts/Type1/"
FontPath "/usr/X386/lib/X11/fonts/Speedo/"
FontPath "/usr/X386/lib/X11/fonts/75dpi/"
FontPath "/usr/X386/lib/X11/fonts/100dpi/"
FontPath    "/usr/X386/lib/X11/fonts/chinese/"
#
# Uncomment this to cause a core dump at the spot where a signal is
# received. This may leave the console in an unusable state, but may
# provide a better stack trace in the core dump to aid in debugging
#
# 是不是允許產生core dump

```

```

# NoTrapSignals

#
*****

# Input devices 輸入設備

#
*****

#
# Enable this to use the XQUEUE driver for keyboard and mouse handling
# under System V. This may go away in the future.
#
# Note - If you use XQUEUE, you must comment out the keyboard and
#      mouse definitions.
# 是否使用XQUEUE驅動程式
# Xqueue
#
# Keyboard and various keyboard-related parameters
#
# 鍵盤的設定，依原來設定的即可
Keyboard
    AutoRepeat 500 5
    ServerNumLock
# Xleds    1 2 3
# DontZap
#
# To set the LeftAlt to Meta, RightAlt key to ModeShift,
# RightCtl key to Compose, and ScrollLock key to ModeLock:
#
# LeftAlt  Meta

```

```

# RightAlt  ModeShift
# RightCtl  Compose
# ScrollLock ModeLock
#
# Mouse definition and related parameters
#
# 設定滑鼠型態與對應的設備檔
Logitech    "/dev/sio00"
# BaudRate  9600
# SampleRate 150
# Emulate3Buttons
#
*****
# Graphics drivers 圖型驅動程式
#
*****
# 記錄影像硬體設備
#
# The 8-bit colour SVGA driver
#
# 被XF86_SVGA所讀取的8位元彩色Super-VGA
VGA256
#
# To disable SpeedUp, use NoSpeedUp
#
# NoSpeedUp
# Virtual 1152 900
Virtual    1024 768

```

```

ViewPort  0 0
Modes     "1024x768i" "800x600" "640x480"
#
# The 16-colour VGA driver
#
# 被XF86_VGA16所讀取的VGA
VGA16
Virtual   800 600
ViewPort  0 0
Modes     "640x480"
#
# The 1-bit mono SVGA driver
#
# 被XF86_Mono所讀取的1位元單色VGA
VGA2
Virtual   800 600
ViewPort  0 0
Modes     "640x480"
# Modes   "800x600" "640x480"
#
# The Hercules driver. For Hercules, the only valid configuration option
# is ScreenNo (refer to the manual page).
#
# 被XF86_Mono所讀取的Hercules
# HGA2
#
# The alternate monochrome driver. Refer to the XF86_Mono manual page.
#

```



```

# 被XF86_Mono所讀取的單色設備
# BDM2
#
# The accelerated servers (S3, Mach32, Mach8, 8514)
#
# 被XF86_S3、XF86_Mach32、XF86_Mach8、與XF86_8514所讀取的加速卡設備
# ACCEL
#
# Virtual    1024 768
# Viewport   0 0
# Modes      "1024x768" "800x600" "640x480"
#
# For boards with a programmable clock generator, you use a line like:
#
# Clocks "icd2061a"
#
*****
# Database of video modes 影像模式的資料庫
#
*****
# 構建顯示器模式
ModeDB
# name      clock horizontal timing  vertical timing  flags
"640x480"   25    640 664 760 800    480 491 493 525
           31    640 664 704 832    480 489 492 520
"800x600"   36    800 824 896 1024   600 601 603 625
           40    800 840 968 1056   600 601 605 628
           50    800 856 976 1040   600 637 643 666

```

```

"1024x768i" 44 1024 1040 1216 1264 768 777 785 817 Interlace
#"1024x768i" 45 1024 1056 1256 1288 768 768 775 807 Interlace
#"1024x768" 65 1024 1032 1176 1344 768 771 777 806
# 75 1024 1048 1184 1328 768 771 777 806
# 85 1024 1032 1152 1360 768 784 787 823
"1280x1024i" 80 1280 1296 1512 1568 1024 1025 1037 1165 Interlace
"1280x1024" 110 1280 1328 1512 1712 1024 1025 1028 1054
135 1280 1312 1456 1712 1024 1027 1030 1064
#
# Refer to README.Config, modeDB.txt, and VideoModes.doc for actual
# modes, and information on how to calculate and adjust them.
#

```

5.1.4 測試：

當安裝完成之後，將機器重新開機，以使修改過的設備檔讓系統接受。開機後，鍵入 `startx` 便可啓動 XFree86。亦可以用 `startx >& startx.log` 來將錯誤訊息存入 `startx.log` 檔案內。當啓動 XFree86 時，會依 `/usr/X386/lib/X11/xinit/xinitrc` 檔案的設定執行預設的應用程式。使用者在自己的目錄內建立 `.xinitrc` 檔，以自行設定啓動時的應用程式。由於 XFree86 預設的視窗管理程式為 `twm`，若要離開 X 視窗，需在主控台 (`console xterm`) 視窗內鍵入 `exit`，或是按 `Ctrl+Alt+BackSpace` 關閉 X 伺服器。另外，按 `Ctrl+Alt++` 可切換視訊模式。樣本機器的執行結果正常，只是畫面的顯示還差強人意，這是因為預設的 `twm` 視窗管理程式並不是很美觀。

5.1.5 討論與結論：

X 視窗一直是在視窗標準化的過程中，佔一席重要地位。因為其開放的架構，可攜性強等優點，幾乎受所有電腦廠商的青睞。所以未來 X 視窗將是整合多種能力的視窗系統，以目前在 UNIX 上的視窗系統而言，X 視窗幾乎是視窗使用者不二的選擇。

5、2 節 FVWM 之建置

5.2.1 前言：

XFree86提供我們視窗圖型環境，方便我們使用電腦，增加對電腦的親和力；然而，好的視窗圖型環境還需要有好的視窗管理程式來輔助，才能幫我們改善整個螢幕的美感，使畫面看起來更舒服。

5.2.2 簡介：

fvwm是一個輕量級的視窗管理程式，提供一個簡單的虛擬桌面(virtual desktop)，一個立體(3-D)的視窗裝飾，以及彩色的圖像(icon)。fvwm對記憶體消耗量約為twm的二分之一至三分之二。

fvwm的目標在於小記憶體的機器，不容易使用滑鼠的膝上型電腦(lapton)，以及偏好Motif視窗管理程式的使用者。所有的視窗功能皆能在沒有滑鼠的情況下完成。

5.2.3 安裝說明：

5.2.3.1 軟體需求

樣本機器安裝的作業系統為FreeBSD，X視窗是XFree86版本2.0。

5.2.3.2 抓取檔案

在中央大學的FTP站softu1.ncu.edu.tw中，目錄/pub4/FreeBSD/packages內有fvwm-0.99_bin.tgz檔案，此為fvwm版本0.99的可執行程式(binaries)之壓縮檔。

5.2.3.3 處理壓縮檔案

利用gunzip與tar程式可將fvwm-0.99_bin.tgz解壓還原，如下：

```
cat fvwm-0.99_bin.tgz | gunzip | tar xvf -
```

執行上述命令後，會自動在目前目錄產生以下檔案：

+COMMENT	- fvwm的簡介
+CONTENTS	- fvwm的安裝步驟
+DESC	- fvwm的說明檔
fvwm	- fvwm的執行程式
fvwm.1	- fvwm的線上輔助檔(man file)
system.fvwmrc	- fvwm的執行命令檔(run commands file)

其中，+COMMENT的內容為：

Fvwm-0.99: A relatively small virtual window manager with a Motif look.

而+DESC的內容為：

Fvwm-0.99: A relatively small window manager which provides a 3-D look for window decorations and a virtual desktop. It is reconfigurable in a manner similar to twm. Most of the Twm functionality has been maintained, with some new features added.

5.2.3.4 安裝

將fvwm置於/usr/X386/bin下，並更改其檔案屬性(file mode)為555(即-r-xr-xr-x)，再將fvwm.1置於/usr/X386/man/man1下，改其檔案屬性為444(即-r--r--r--)。執行mkdir /usr/X386/lib/X11/fvwm，建立fvwm的執行命令目錄，然後將system.fvwmrc放入，如此便完成安裝。

5.2.3.5 修改設定檔

為了配合需求，可以修改系統的system.fvwmrc檔案。樣本機器修改後的內容如下：

```
# The definition of menu Window Ops2 has the menu hot-key key-letters
# tagged. If you don't use menu-hot-keys, you should remove the
# &'s in that menu.

# Also, B&W icons are assumed. If you have color icons configured, you
# might as well change it now.

#####

# set up the colors 設定顏色

# this is used for non-selected windows, menus, and the panner
StdForeColor      Black
StdBackColor      #60a0c0

# this is used for the selected window
HiForeColor       Black
HiBackColor       #c06077
PagerBackColor    #5c54c0
```

```

PagerForeColor    orchid
#####
# Now the fonts - one for menus/icons, another for window titles
# 設定螢幕顯示字形
Font              -adobe-helvetica-medium-r-*-*-120-*-*-*-*-*
WindowFont        -adobe-helvetica-bold-r-*-*-120-*-*-*-*-*
IconFont          -adobe-helvetica-medium-r-*-*-100-*-*-*-*-*
# PagerFont is used to mark windows in the Pager with their icon names.
# This is really only useful for DeskTopScale < 20 or so
PagerFont         5x8
#####
# Set up the major operating modes 設定主要操作模式
#
# Set windows to auto-raise after 750 milliseconds if you like it.
# Autoraise can sometimes obscure pop-up windows. Performance is now
# similar to olvwm's auto-raise feature.
#AutoRaise 750
# Normally, we'll be in focus-follows mouse mode, but uncomment this
# for mwm-style click-to-focus
#ClickToFocus
# Auto Place Icons is a nice feature...
# This creates two icon boxes, one on the left side, then one on the
# bottom. Leaves room in the upper left for my clock and xbiff,
# room on the bottom for the Pager.
IconBox -20 1 -1 -50
# If you uncomment this, and make sure that the WindowList is bound to
# something, it works pretty much like an icon manager.
#SuppressIcons

```

```

# If you don't like the default 150 msec click delay for the complex functions
# change this and uncomment it.
#ClickTime 150
#####
# Set up the virtual desktop and pager 設定虛擬桌面與pager
#set the desk top size in units of physical screen size
DeskTopSize 3x3
# and the reduction scale used for the panner/pager
DeskTopScale 32
# Use the Fvwm Pager
Pager -5 -5
#The pager is automatically sticky, but I like it to StayOnTop
StaysOnTop Fvwm Pager
#The pager is automatically sticky, but I like it without a title
NoTitle Fvwm Pager
#Might as well leave it out of the window list too...
WindowListSkip Fvwm Pager
# flip by whole pages on the edge of the screen.
EdgeScroll 100 100
# A modest delay before flipping pages seems to be nice...
# I thresh in a 50 pixel Move-resistance too, just so people
# can try it out.
EdgeResistance 250 50
#####
# list the windows that don't want a title bar 設定不須顯示標題的視窗
NoTitle xclock
NoTitle xdclock
NoTitle Clock

```

```

NoTitle xbiff
#####
# Here's where I could list NoBorder windows that don't get frames.
# 設定不須顯示外框的視窗
#NoBorder xterm
#####
# now the windows that "Stick to the glass", ie don't scroll with the desktop
# 設定不隨桌面捲動的視窗
Sticky xclock
Sticky xdclock
Sticky oclock
Sticky Clock
Sticky xbiff
#####
# now list windows that want to "stay on top," ie stay visible until explicitly lowered
# 設定保持可見的視窗
#StaysOnTop xclock
#####
# now list windows that don't need to show up in the window list
# 設定不出現在視窗列表的視窗
WindowListSkip Clock
WindowListSkip xbiff
WindowListSkip xclock
#####
# define some common icon bitmaps 定義一般的圖像二元圖
# These are the color icons.
# Of course, you need to change the directory to the one where they
# really live.

```

```

#Icon "xterm"    /usr/include/X11/pixmaps/xterm.xpm
# Here are some monochrome icon bitmaps that everyone should have
Icon "rxvt"     /usr/include/X11/bitmaps/terminal
Icon "xterm"    /usr/include/X11/bitmaps/Term
# I have included a default.xbm in the fvwm_icons subdirectory
Icon ""         /usr/include/X11/bitmaps/default.xbm
#####
#now define the menus - defer bindings until later 定義選單項目
# This menu is invoked as a sub-menu - it allows you to quit, 離開時選項
# restart, or switch to another WM.
Popup "Quit-Verify"
    Title "Really Quit Fvwm?"
    Quit "Yes, Really Quit"
    Restart "Restart Fvwm" fvwm
    Restart "Start twm" twm
    Restart "Start mwm" mwm
    Restart "Start olwm" /usr/openwin/bin/olwm
    Nop ""
    Nop "No, Don't Quit"
EndPopup
# This menu will fire up some very common utilities 一般工具選項
Popup "Utilities"
    Title    "Utilities"
    Exec    "Xterm"      exec xterm &
    Exec    "Rxvt"       exec rxvt &
    Exec    "Top"        exec rxvt -T Top -n Top -e top &
    Exec    "Calculator" exec xcalc &
    Nop ""

```



```

    Popup "Exit Fvwm" Quit-Verify
    Nop ""
    Refresh "Refresh Screen"

EndPopup
# This defines the most common window operations 視窗操作選項

Popup "Window Ops"
    Title "Window Ops"
    Move "Move"
    Resize "Resize"
    Raise "Raise"
    Lower "Lower"
    Iconify "(De)Iconify"
    Stick "(Un)Stick"
    Maximize "(Un)Maximize"
    Maximize "(Un)Maximize Vertical" 0 100
    Nop ""
    Destroy "Destroy"
    Delete "Delete"

EndPopup
# this is just like window-ops, but it has the title
# omitted, and menu-hot-keys keyletters tagged

Popup "Window Ops2"
    Move "&Move"
    Resize "Re&size"
    Raise "&Raise"
    Lower "&Lower"
    Iconify "(De)&Iconify"
    Stick "(Un)&Stick"

```

Maximize "(Un)Ma&ximize"

Maximize "(Un)Maximize &Vertical" 0 100

Nop ""

Destroy "&Destroy"

Delete "Delete"

EndPopup

#####

Now define some handy complex functions 定義一些複雜功能

This one moves and then raises the window if you drag the mouse,

only raises the window if you click, or does a RaiseLower if you double click

Function "Move-or-Raise"

Move "Motion"

Raise "Motion"

Raise "Click"

RaiseLower "DoubleClick"

EndFunction

This one pops up window-ops2 in a sticky manner, that is

if you just click, the menu stays up, as opposed to the

default behavior, which is to disappear on a simple click

Function "window_ops_func"

PopUp "Click" Window Ops2

PopUp "Motion" Window Ops2

EndFunction

This function is used to over-load the maximize button, so that

it maximizes vertically for a click, but does full maximization for

a double-click

Function "maximize_func"

Maximize "Motion" 0 100

```

    Maximize    "Click" 0 100
    Maximize    "DoubleClick" 100 100
EndFunction
# This one moves and then lowers the window if you drag the mouse,
# only lowers the window if you click, or does a RaiseLower if you double
# click
Function "Move-or-Lower"
    Move        "Motion"
    Lower       "Motion"
    Lower       "Click"
    RaiseLower  "DoubleClick"
EndFunction
# This one resizes and then raises the window if you drag the mouse,
# only raises the window if you click, or does a RaiseLower if you double click
Function "Resize-or-Raise"
    Resize      "Motion"
    Raise       "Motion"
    Raise       "Click"
    RaiseLower  "DoubleClick"
EndFunction
#####
# This defines the mouse bindings 定義滑鼠按鍵功能
# First, for the mouse in the root window
# Button 1 gives the Utilities menu
# Button 2 gives the Window Ops menu
# Button 3 gives the WindowList (like TwmWindows)
# I use the AnyModifier option for the modifier field, so you can hold down
# any shift-control-whatever combination you want!

```

```

# Button Context Modifi Function
Mouse 1 R A PopUp "Utilities"
Mouse 2 R A PopUp "Window Ops"
Mouse 3 R A WindowList

# Now the title bar buttons

# Any button in the left title-bar button gives the window ops menu
# Any button in the right title-bar button Iconifies the window
# Note the use of "Mouse 0" for AnyButton.

# Button Context Modifi Function
Mouse 0 1 A Function "window_ops_func"
Mouse 0 2 A Iconify
Mouse 0 4 A Function "maximize_func"

# Now the rest of the frame

# Here I invoke my complex functions for Move-or-lower, Move-or-raise,
# and Resize-or-Raise.

# Button 1 in the corner pieces, with any modifiers, gives resize or raise
Mouse 1 F A Function "Resize-or-Raise"
# Button 1 in the title, sides, or icon, w/ any modifiers, gives move or raise
Mouse 1 TSI A Function "Move-or-Raise"
# Button 2 in an icon, w/ any modifiers, gives de-iconifies
Mouse 2 I A Iconify
# Button 2 in the corners, sides, or title-bar gives the window ops menu
Mouse 2 FST A PopUp "Window Ops"
# Button 3 anywhere in the decoration (except the title-bar buttons)
# does a move-or-lower
#Mouse 3 TSIF A Function "Move-or-Lower"
Mouse 3 TSIF A RaiseLower
Mouse 1 A M RaiseLower

```

#####

Now some keyboard shortcuts. 定義鍵盤快速選項

Arrow Keys

press arrow + control anywhere, and scroll by 1 page

Key Left A C Scroll -100 0

Key Right A C Scroll +100 +0

Key Up A C Scroll +0 -100

Key Down A C Scroll +0 +100

press arrow + meta key, and scroll by 1/10 of a page

Key Left A M Scroll -10 +0

Key Right A M Scroll +10 +0

Key Up A M Scroll +0 -10

Key Down A M Scroll +0 +10

press shift arrow + control anywhere, and move the pointer by 1% of a page

Key Left A SC CursorMove -1 0

Key Right A SC CursorMove +1 +0

Key Up A SC CursorMove +0 -1

Key Down A SC CursorMove +0 +1

press shift arrow + meta key, and move the pointer by 1/10 of a page

Key Left A SM CursorMove -10 +0

Key Right A SM CursorMove +10 +0

Key Up A SM CursorMove +0 -10

Key Down A SM CursorMove +0 +10

Keyboard accelerators

Key F1 A M Popup "Utilities"

Key F1 A M5 Popup "Utilities"

Key F2 A M Popup "Window Ops"

Key F3 A M WindowList

```

Key F4      A  M  Iconify
Key F5      A  M  Move
Key F6      A  M  Resize
Key F7      A  M  CirculateUp
Key F8      A  M  CirculateDown

```

#Page Up/Dage Down keys are used to scroll by one desktop page

in any context, press page up/down + control

in root context, just pressing page up/down is OK

#

I prefer the non-wrapping scroll. These are for example purposes only

```
#Key Next  A  C  Scroll 100000 0
```

```
#Key Next  R  N  Scroll 100000 0
```

```
#Key Prior A  C  Scroll -100000 0
```

```
#Key Prior R  N  Scroll -100000 0
```

此外，還必須修改/usr/X386/lib/X11/xinit/xinitrc的內容，加入tvwm &的敘述。若該檔案內原有執行其它視窗管理程式(如twm或olwm)的敘述，必須將其刪除或標示為註解。

5.2.4 測試：

執行startx，便可以啓動fvwm，此時樣本機器的表現比執行twm時美觀多了，並且可快速執行外部程式(如CXterm)。每一位使用者還可依自己的喜好，在自己的目錄中(home directory)建立一名為.fvwmrc的隱藏檔，內容定義方式與系統設定檔system.fvwmrc一樣。如此，便可擁有自己與眾不同的視窗型式。此外，若系統的/usr/X386/lib/X11/xinit/xinitrc沒有執行fvwm的敘述，也可在自己目錄中的.xinitrc檔案內設定執行fvwm，方法也與系統設定方法一樣。

5.2.5 討論與結論：

X視窗已經是眾人所公認的視窗標準，但X視窗並未定義使用者圖形介面(GUI)。目前市場上分OSF(Open Software Foundation)的Motif與UI的OpenLook二派。fvwm

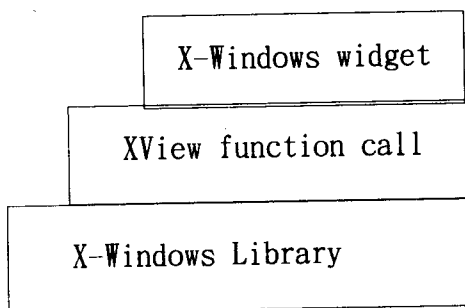
是以Motif介面為基底的視窗管理程式，相信即使是OpenLook的支持者，亦會佩服它的優越特性。

5、3節 XView之建置

5.3.1 前言

什麼是 XView:

XView 是 X-Windows-Based Visual /Integrated Environment for Workstation 的縮寫, 是一個在 X-Windows 下執行的交談式/圖形化使用者介面撰寫程式。XView 也提供了撰寫 X-Windows 應用程式所需的函式。這些物件遵循 OPEN-LOOK 圖形化使用者介面的標準。使用 XView 可以節省發展 X-Windows 應用程式的時間,對於一些如 toolbar , 對話方塊等等使用者介面的撰寫,可以省下查 X-Windows Library call 的時間 , 直接用 XView 的 function call 便可。下圖代表 XView與X-Windows Library 的關係。



5.3.2 XView 系統需求:

XView 需要在 ICCCM-compliant windows manager 下執行, 例如:olwn (openlook Window Manager), 而且 , 由於 XView是建構在 X-Windows Library 之上的, 所以也必須要有完整的 ICCCM Xlib 的支援。此次之系統是架構在X-Windows Version 11 Release 5 (X11R5)之上, 若要在 R5 之前的版本下執行, 則要參考 /config/XView.cf 定義 PRE_R4_ICCCM 的旗標, 設定 .Xdefaults 中的選項 xvview.icccmcompliant 之值為 FALSE 。

XView 的 package 中包含了以下的子目錄：

lib/libxview	the XView tool kit
lib/libolgx	the OLGX tool kit
clients/clock	XView 的時鐘
clients/cmdtool	XView 終端機模擬程式
clients/olwm	the OPENLOOK windows manager
clients/olwmslave	olwm 的 helper
contrib/examples	XView 的一些 example
fonts/bdf	給 PRE_R4 server 的 OPENLOOK 字形
doc	有關 XView 的文件
misc/message_files	Localization 的訊息
misc/xrdb_files	X 的 source file for olwm/Xview3.2

解壓的檔案大小大約為 20 Mega, 在共用 lib 的狀況下要多使用 18 Mega 來安裝, 安裝後的大小約為 8.5 Mega。

5.3.3 來源:

XView 的來源是在 `softu1.ncu.edu.tw` 中 `/pub4/XFree86/package` 下的 `xview32s.tgz`(source code), `xview32l.tgz`(library files)。同一個目錄下還有一個已在 FreeBSD 下 compile 完成的 `xview32b.tgz` (binary file)。此外, 在其它的地方如: `NCTUCCCA.edu.tw` 及 `ftp.csie.nctu.edu.tw` 下都可以找到相同的檔案。以

```
cat xview32?.tgz | gunzip | tar xvf -
```

 便可以把這些檔打開。

5.3.4 安裝(以 FreeBSD 為準):

由於安裝 XView 時會同時更新系統原有的 X-Windows 的 Library, 所以需要 root 的權限來做 XView 的安裝工作。

在安裝 XView 前要先設定你的 X-windows 的目錄, 在 csh 下要做:

```
%setenv OPENWINHOME [X-windows home-directory]
```


例如:

```
%setenv OPENWINHOME /usr/X386
```

在 sh 下則為:

```
$OPENWINHOME=/usr/X385;export OPENWINHOME
```

接著, 在 xview3.2 的子目錄下執行

```
%xmkmf
```

系統便會把系統本身的設定值和 Imakefile 合成新的 Makefile, 並把原本的 Makefile 改名為 Makefile.bak

此外, 如果你在 /xview3.2/config.XView.cf 中, 定義 XvI18Level 為 4 (內定值為 3)則要修改 Makefile 中的

```
XVCLIENTI18NDEFINES= -DOW_I18NL4 -DOW_I18N_L3-    DOW_I18N  
-DFULL_R5
```

改為

```
XVCLIENTI18NDEFINES= -DOW_I18N_L3 -DOW_I18N -DFULL_R5
```

完成設定後, 進入 xview 的子目錄, 再鍵入

```
%make
```

如果沒有 error message 或程式被中斷, 便完成安裝了。

5.3.5 錯誤訊息及修正方式:

1. "*Can't find include file XView.tmpl.*": 若出現此訊息, 則要修改IMAKEINCLUDE 的值, 在 csh 下是

```
%setenv IMAKEINCLUDE -I<X11>/lib/X11/config -I<XView>/config
```

若為 sh, 則是

```
$IMAKEINCLUDE="-I<X11>/lib/X11/config -I<XView>/config"
```

`$export IMAKEINCLUDE`

2. "*too many open file*": 系統設定可開啓檔案數太小, 在 `cs` 下以
`%limit openfiles 200` 來修正。

5、4 節 Ghostscript 及 Ghostview 之建置

5.4.1 前言：

What is ghostscript ?

- An interpreter for the ghostscript language , which very closely resembles the PostScript language .
- An interpreter of C procedures (the Ghostscript library) that implement the graphics capabilities that appear as primitive operations in the Ghostscript language .

它是一套用來檢視 *.ps 此類格式的檔案, 並提供轉換, 列印等功能的工具此類檔案可以是具備圖形及文字的檔案。Ghostscript 可以顯示圖形及文字這種格式的檔案較一般格式的文件要大許多(如 pe3 的 text 檔, amipro 的 sam 檔, Word 的 doc 檔 ...)。但是其優點是提供了在 X-Windows 下閱讀具有中文及圖形的檔案能力, 而 Word, Amipro 等文書處理下所編輯的文件亦可轉換成 Ghostscript 格式的檔案, 並用 Ghostscript 來閱讀轉換後文件。Ghostscript 可以在 IBM PC, UNIX System using X-WINDOWS versional, release3,4 and 5 .VAX,VMS with X11R3/4/5 等平台上執行, 而此示範是安裝在 FreeBSD X11R5 上。

而 Ghostview 則是一個架構在 Ghostscript 上的應用程式, 它為 ghostscript 提供一個良好界面方便使用者開啓, 檢視, 列印檔案, 使用者可以用滑鼠直接選取各項功能, 不必像在 Ghostscript 下必須由使用者自行輸入命令, 其中更包括了局部放大, 直接設定紙張大小等功能, 使使用者方便使用 Ghostscript。

5.4.2 GhostScript 與 GhostView 來源：

此次安裝的版本是 Ghostscript 2.6.1, 總共需要兩個檔案：

來源：nctucca.edu.tw

目錄：/UNIX/gnu

檔名：ghostscript-2.6.1.tar.gz

ghostscript2.6.1-fonts.tar.gz

此次安裝的版本是 GhostView-1.4 檔案共一個：

節點：nctucca.edu.tw

目錄：/UNIX/gnu

檔案：ghostview-1.4.tar.gz

5.4.3 GhostScript 安裝過程：

(1) 解開 ghostscript-2.6.1.tar.gz 檔

```
gunzip ghostscript-2.6.1.tar.gz
```

```
tar -xvpf ghostscript-2.6.1.tar
```

(2) 執行 `ln -s unix-cc.mak makefile`

===== 以下僅供參考 start =====

(3) 修改 makefile 檔中

```
XINCLUDE=-I/usr/X386/include
```

```
XLIBDIRS=-L/usr/X386/lib
```

(4) 修改 x_.h 檔中

```
#include <stddef.h>
```

```
#include </usr/X386/include/X11/Xlib.h
```

```
#include </usr/X386/include/X11/Xatom.h
```

```
#include </usr/X386/include/X11/Xutil.h
```

```
#include </usr/X386/include/X11/Intrinsic.h
```

```
#include </usr/X386/include/X11/StringDefs.h
```

```
#include </usr/X386/include/X11/Shell.h
```

===== End =====

(5) 執行 `make install`

產生兩個目錄：`/usr/local/lib/ghostscript`

`/usr/local/bin`

(可以在 `makefile` 中自定)

(6) 解開 `ghostscript-2.6.1-fonts.tar.gz` 檔

`gunzip ghostscript-2.6.1-fonts.tar.gz`

`tar -xvpf ghostscript-2.6.1-fonts.tar`

將解開檔案放在 `/usr/local/lib/ghostscript/fonts` 下

如此便完成安裝，其中 `/usr/local/bin` 存放執行檔 `gs` 另外還有 `bdf tops`, `gsdj`, `gslp`, `ps2ascii`, `font2c`, `gsbi`, `gsnd`, `ps2epsi` 八個檔。而在另外一個子目錄 `/usr/local/lib/ghostscript` 中則存放許多 `*.ps` 檔，為範例檔，而字形檔則存放在 `/usr/local/lib/ghostscript/fonts`，供不同字形檔所需，若所需字形沒有，則文件

會呈現空白。

5.4.4 GhostView 工作原理：

`Ghostview` 實際上是透過 `Ghostscript` 來轉換 `*ps` 檔的，它本身只是一個界面，一個方便使用者使用 `Ghostscript` 的界面，當執行 `GhostView` 時，`GhostView` 會自己自動執行 `GhostScript` 來完成所欲執行的工作。

5.4.5 GhostView 安裝過程：

(1) 執行 `xmkmf`

此指令會做下列動作：

`mv Makefile Makefile.bak`

```
imake -PUseInstalled -I/usr/X386/lib/X11/config
```

```
# ( /usr/X386/bin/imake )
```

依系統環境變數適當修改 makefile

執行後加入的參數：

```
BOOTSTRAPCFLAGS = DFreeBSD
```

```
    NROFF = nroff
```

```
    NEQN = neqn
```

```
STD_CPP_DEFINES = -traditional -DFreeBSD
```

```
    STD_DEFINES = DFreeBSD
```

```
EXTRA_LOAD_FLAGS =
```

```
EXTRA_LIBRARIES = -lgnumalloc
```

```
    OS_LIBRARIES =
```

```
INSTKMEMFLAGS = -m 2755 -g kmem
```

執行後修改後的參數：

```
PROJECTROOT = /usr/X386
```

```
CCOPTIONS = -m486
```

```
LDCOMBINEFLAGS = -X -r
```

```
MACROFILE = X386
```

```
USRLIBDIR = /usr/X386/lib
```

```
    BINDIR = /usr/X386/lib
```

```
    INCROOT = /usr/X386/include
```

```
    MANPATH = /usr/X386/man
```

```
FILEMANSUFFIX = 5x
```

(2) 執行 make install

此指令會做下列動作：

```
sh ad2c Ghostview.ad > app defaults.h
```

```

# ( Imakefile 中一行指令 )

gcc -O2 -m486 -I/usr/X386/include -DfreeBSD

gcc .....

gcc .....

.

.

.

install -c -s ghostview /usr/X386/bin

install -c -m 0444 Ghostview.ad /usr/X386/lib/X11

                                /app-defaults/Ghostview

install in .done

```

make install 會 copy Ghostview.ad 到 /usr/X386/lib/X11/app-defaults 下，檔名為 Ghostview，這個檔是 GhostView 執行時呼叫 gs 所必需讀取的，當 GhostScript 有更改版本時，可以在此檔中指定欲使用的新版本，加入一行 Ghostview.interpreter /ghostscript-new <--- (新版所在目錄)

5.4.6 使用方法:

執行 ghostview 只要鍵入 ghostview 即可進入視窗，或鍵入ghostview file直接指定檔案，然後就會開啓 Ghostview 視窗。

5.4.7 如何將一般文字檔轉換成 Ghostscript 格式檔：

當用一般的文書處理編輯檔案後，如：Word，Amipro，可以借助 windows 所提供的 postscript 印表機來幫我們轉換成 Ghostscript 格式的檔案，其方式如下：

- (1) 驅動 postscript 印表機。
- (2) 在列印下選擇 postscript 印表機。
- (3) 在設定下選擇選項中的列印至 EPS 檔。

5、5節 Hytelnet之建置

5.5.1 前言：

Hytelnet 是一套查詢系統，提供世界上各節點 address 包括 BBS，gopher，WWW server，hytelnet server 並可直接讓使用者連線過去，操作簡單，在交大的nctuccca.edu.tw 有 hytelnet server，使用者只要 telnet nctuccca.edu.tw，然後再用 hytelnet 為 login name，就可以進入 hytelnet。hytelnet 本身就是一個索引(index)，它記錄了各節點 address，並加以分類，然後依分類的類別記錄在各分類檔。在此藉由修改 hytelnet 來製作一簡易線上輔助系統；但是 hytelnet 本身並不能顯示中文，所以必須更改成可顯示中文，來達成建立一套中文化的簡易線上輔助系統之目標。

5.5.2 來源：

address : 140.116.2.11
directory : /pub/unix/hytelnet
filename : hytelnet.tar.Z

5.5.3 安裝：

先把 hytelnet.tar.Z 解開：

```
uncompress hytelnet.tar.Z  
tar -xvpf hytelnet.tar
```

解開後有以下檔案：

```
-rw-r--r--  1 easter   286 Dec  3 1992 Makefile  
-rw-r--r--  1 easter  3087 Jan 27 1993 README.UNIX  
-rw-rw-rw-  1 easter   836 Jan 22 1992 help  
  
-rw-r--r--  1 easter   2931 Oct 22 1992 hytelnet.1  
-rw-r--r--  1 easter  23914 Oct 22 1992 hytelnet.c
```

```
-rw-rw-r-- 1 easter 51200 Mar 11 01:23 hytelnet.tar
-rw-r--r-- 1 easter 2493 Jun 21 1993 read.me
-rw-r--r-- 1 easter 1004 Jun 21 1993 start
-rw-r--r-- 1 easter 673 Jun 21 1993 whatis
```

再執行 `make` :

```
make hytelnet.c
```

執行過程 :

```
cc -c hytelnet.c
cc -o hytelnet hytelnet.o -lcurses -ltermcap
```

`Hytelnet` 執行時會抓取 `start` 此檔案，顯示其內容並將檔案內 `<>` 兩符號中字符串反白當做選項。

- (1) 反白字符串若為一資料檔名稱則顯示其內容於螢幕上。
- (2) 若此目錄沒有這檔案，則取反白字符串前三字元為目錄名稱並進入此目錄，然後再抓取以此反白字符串為名稱的檔案並顯示其內容於螢幕上，通常這個檔為一分類檔（如 `start file` 即為一分類檔，提供使用者多項選擇）。
- (3) 分類檔的選項跟檔案所在的目錄名稱有關，即 `<>` 中的檔名前三字元必須是所在目錄的名稱。

舉例來解釋：

- a. 在 start 檔中有提供 `ghostscript & ghostview <ghost>`
此選項，當使用者在 `<ghost>` 這 highlight 選項按 enter
程式就在 `hytelnet` 這目錄找 `ghost` 這個檔案，當程式找不到
`ghost` 檔，就進入 `gho` 此目錄（取其前三字元），然後
在 `gho` 目錄中找 `ghost` 檔，並 `dump ghost` 檔。

- b. 顯示 `ghost` 檔後（`ghost` 必為一分類檔），繼續選擇選項，
`ghost` 檔中應提供 `gho` 此目錄中所有資料檔選項，其中有：

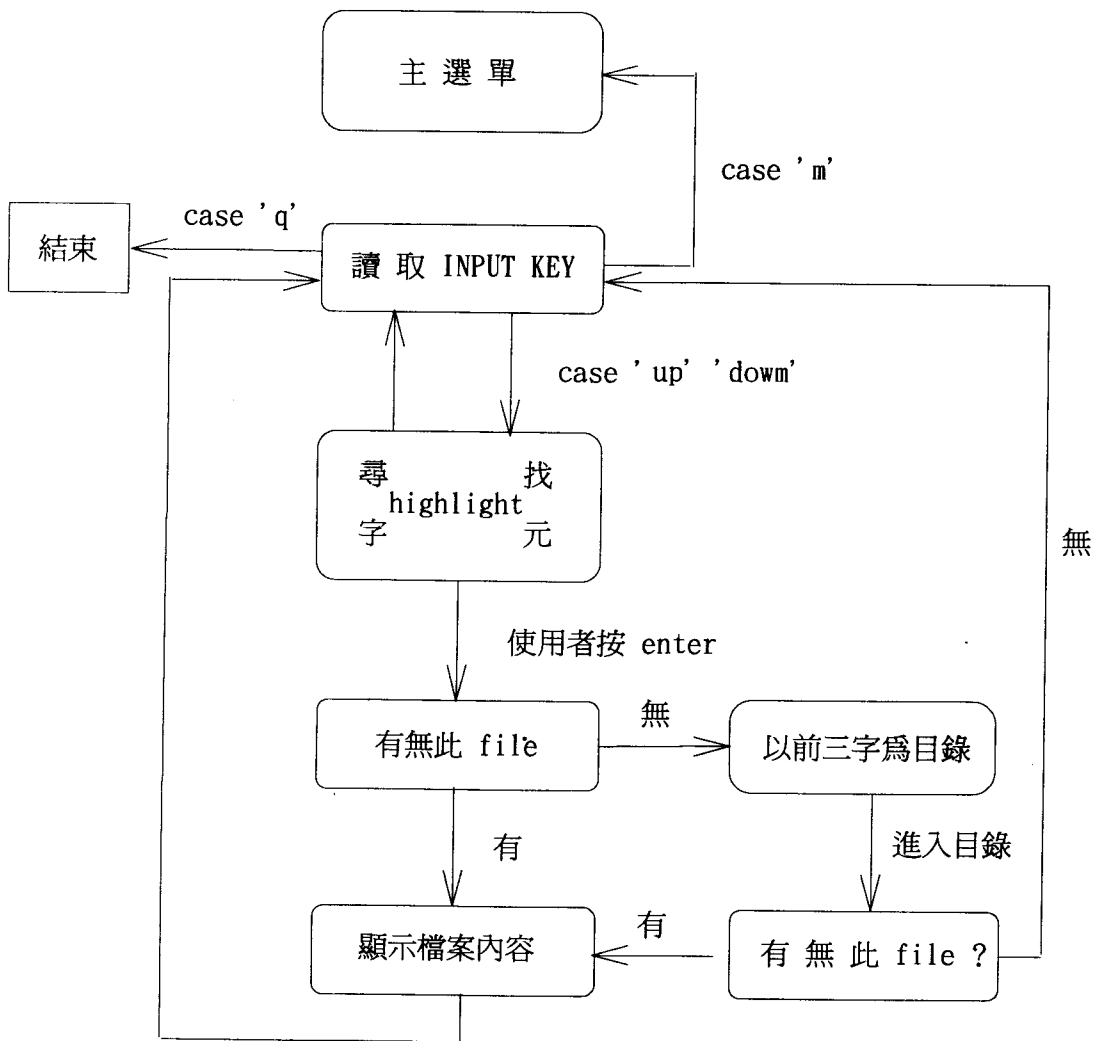
`Ghostscript-2.6.1 相關資料 <Gho1>`

`Ghostview-1.5 相關資料 <Gho2>`

兩項，注意，`<>` 中檔名前三字元為 `gho`，否則會找不到！

- c. 另一個要注意的是 `<>` 中不可以有空白即不可以是 `<gho1 >`
而 `<>` 中的字母大小寫則無關緊要，程式會自動轉換，即
`<ghost>`，`<GHOST>` 或 `<Ghost>` 都可以，但是資料檔實際
檔名必須為小寫，不可以是大寫；否則會找不到檔案。

* 以下是程式的流程圖（圖一）：



(圖一)

修改部份：

原程式不能顯示中文是因為其中參數 `printable` 原因，故將其值改定為永遠 `true`，如下：

原程式為：

```
#define printable(c) (((c)>31 && (c)<=127) || (c)==9 || (c)==10)
```

修改成爲：

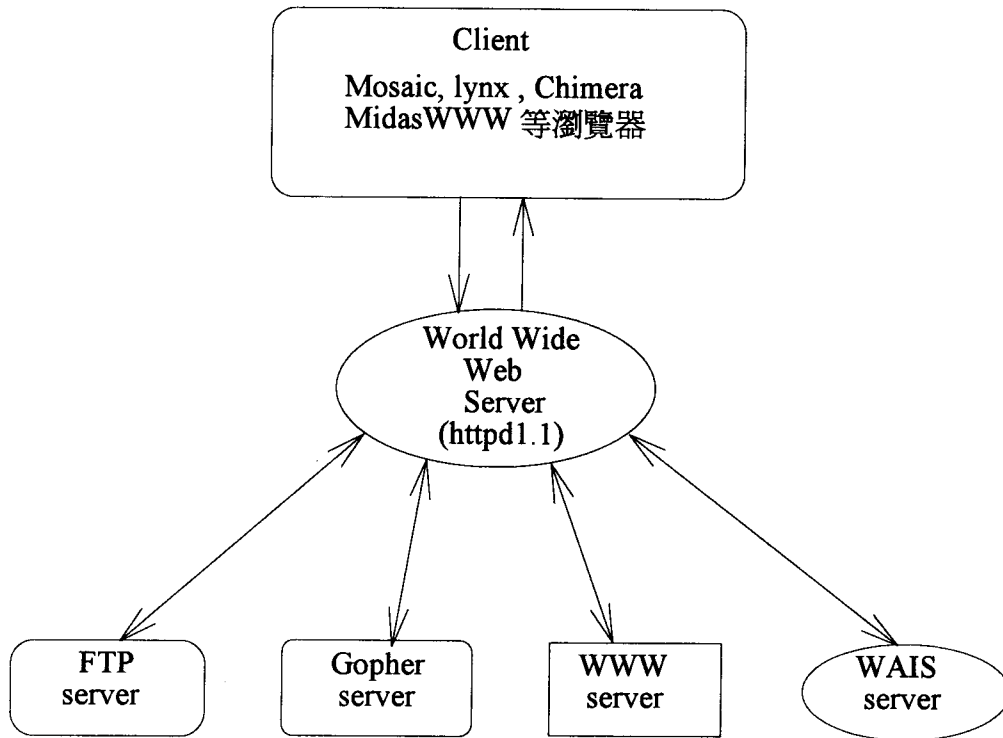
```
#define printable(c) 1
```

這樣便可以顯示中文，但亦犧牲掉一些功能，即資料檔必需是 `ascii` 型式而不能是 `binary` 型式；否則 `LF` 記號會被顯示出來。

5、6 節 Http之建置

5.6.1 前言:什麼是HTTPD

在了解何謂HTTPD前,我們要先知道什麼是WWW,所謂的WWW是World-Wide Web的縮寫,這個計劃是由CERN(the European Laboratory for Particle Physics)開始推展的,其目的在於建立一個分散式的超媒體系統(distributed hypermedia system)。而HTTPD(HyperText Transfer Protocol Daemon)就是這個系統中提供的伺服器(server),能對系統的使用者提供超媒體的文件。關於WWW文件的編輯在附錄中將有詳細的說明,WWW的工作流程如下圖。



5.6.2 系統需求:

建立HTTPD需要一臺UNIX系統的主機做為伺服器(server),此外由於WWW是client-server 模式的架構,所以建立HTTPD要佔走系統的一個port,作為資料傳輸之用,一般使用的port是 80。

5.6.3 來源:

這次HTTPD_1.1的來源是Host nctucca.edu.tw (192.83.166.10)

在 /packages/info-systems/WWW/ncsa/ncsa_httpd/httpd_source.tar.gz

DIRECTORY drwxr-xr-x 2048 bytes 00:00 25 Mar 1994

拿到後將其 `cat httpd_source.tar.gz | gunzip | tar xvf -` 便可將檔案打開,其內容有:

```

README  cgi-src/  httpd.conf  logs/  support/
cgi-bin/  conf/  icons/  src/
  
```

5.6.4 安裝(以FreeBSD 為準):

HTTPD的安裝十分簡單,在 /src 下,我們要先去修改 httpd.h 的定義

```
#define HTTPD_ROOT "/home/wing/httpd_1.1"
```

這一行是對你的 HTTPD 的根目錄加以定義接著,不需要更改 **Makefile**,只要選定你的機器種類鍵入 `make <機器名稱>` 就可以了,以FreeBSD 為例,我們只要在 `/src` 下打 `make ibm` 便完成安裝了, `/cgi-bin` 下是 HTTPD 提供的一些工具,由於是一些 binary檔,在不同的機器上可能不能順利的執行,所以我們要在 `/cgi-src` 下重新將其 compile 一遍,同樣的,鍵入 `make ibm` 。

5.6.5 關於系統的設定:

以下是一些關於 HTTPD 設定檔,位於 `/conf` 下,這些檔十分重要,如果設定不對便無法使用,修改的部份我會加粗並且加上底線及中文解釋。

httpd.conf 設定如下:

```
# This is the main server configuration file. It is best to
# leave the directives in this file in the order they are in, or
# things may not go the way you'd like. See URL http://hoohoo.ncsa.uiuc.edu/
# for instructions.
# Do NOT simply read the instructions in here without understanding
# what they do, if you are unsure consult the online docs. You have been
# warned.
# Rob McCool (comments, questions to httpd@ncsa.uiuc.edu)
# ServerType is either inetd, or standalone.
ServerType standalone

# If you are running from inetd, go to "ServerAdmin".
# Port: The port the standalone listens to. For ports < 1023, you will
# need httpd to be run as root initially.
Port 80
```

```
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
# User/Group: The name (or #number) of the user/group to run httpd as.
User nobody
Group #-1
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.
ServerAdmin you@your.address

# ServerRoot: The directory the server's config, error, and log files
# are kept in
#ServerRoot /usr/local/etc/httpd
#定義 HTTPD 所在目錄
ServerRoot /home/wing/httpd_1.1

# ErrorLog: The location of the error log file. If this does not start
# with /, ServerRoot is prepended to it.
ErrorLog logs/error_log

# TransferLog: The location of the transfer log file. If this does not
# start with /, ServerRoot is prepended to it.
TransferLog logs/access_log

# PidFile: The file the server should log its pid to
PidFile logs/httpd.pid

# ServerName allows you to set a host name which is sent back to clients for
#your server if it's different than the one the program would get (i.e. use
```

#"www" instead of the host's real name).

#ServerName new.host.name

access.conf設定如下:

access.conf: Global access configuration

Online docs at <http://hoohoo.ncsa.uiuc.edu/>

I suggest you consult them; this is important and confusing stuff.

/usr/local/etc/httpd/ should be changed to whatever you set ServerRoot to.

<Directory /home/wing/httpd_1.1/cgi-src>

Options Indexes FollowSymLinks

</Directory>

This should be changed to whatever you set DocumentRoot to.

#定義文件所在目錄

<Directory /home/wing/httpd_1.1/htdocs>

This may also be "None", "All", or any combination of "Indexes",

"Includes", or "FollowSymLinks"

Options Indexes FollowSymLinks

This controls which options the .htaccess files in directories can

override. Can also be "None", or any combination of "Options", "FileInfo",

"AuthConfig", and "Limit"

AllowOverride All

Controls who can get stuff from this server.

srm.conf的設定如下:

With this document, you define the name space that users see of your http
server.

Rob (robm@ncsa.uiuc.edu)

DocumentRoot: The directory out of which you will serve your
documents. By default, all requests are taken from this directory, but
symbolic links and aliases may be used to point to other locations.

#定義文件的根目錄

DocumentRoot /home/wing/httpd_1.1/htdocs

UserDir: The name of the directory which is appended onto a user's home
directory if a ~user request is recieved.

UserDir public_html

DirectoryIndex: Name of the file to use as a pre-written HTML

directory index

DirectoryIndex index.html

FancyIndexing is whether you want fancy directory indexing or standard

<Limit GET>

order allow,deny

allow from all

</Limit>

</Directory>

You may place any other directories you wish to have access

information for after this one.

FancyIndexing on

AddIcon tells the server which icon to show for different files or filename


```

# extensions

AddIcon /icons/text.xbm .html .txt

AddIcon /icons/image.xbm .gif .jpg .xbm .tiff

AddIcon /icons/sound.xbm .au

AddIcon /icons/movie.xbm .mpg

AddIcon /icons/binary.xbm .bin

AddIcon /icons/back.xbm ..

AddIcon /icons/menu.xbm **DIRECTORY**

# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
DefaultIcon /icons/unknown.xbm

# AddDescription allows you to place a short description after a file in
# server-generated indexes.
# Format: AddDescription "description" filename
# ReadmeName is the name of the README file the server will look for by
# default. Format: ReadmeName name
#
# The server will first look for name.html, include it if found, and it will
# then look for name and include it as plaintext if found.

ReadmeName README

# IndexIgnore is a set of filenames which directory indexing should ignore
# Format: IndexIgnore name1 name2...

IndexIgnore /.htaccess ~ #

# AccessFileName: The name of the file to look for in each directory

```

for access control information.

AccessFileName .htaccess

DefaultType is the default MIME type for documents which the server
cannot find the type of from filename extensions.

DefaultType text/plain

AddType allows you to tweak mime.types without actually editing it, or to
make certain files to be certain types.

Format: AddType type/subtype ext1

Redirect allows you to tell clients about documents which used to exist in
your server's namespace, but do not anymore. This allows you to tell the
clients where to look for the relocated document.

Format: Redirect fakename url

Aliases: Add here as many aliases as you need, up to 20. The format is

Alias fakename realname

#定義圖像的位置

Alias /icons/ /home/wing/httpd_1.1/icons/

ScriptAlias: This controls which directories contain server scripts.

Format: ScriptAlias fakename realname

#定義 server script 的位置

ScriptAlias /cgi-bin/ /home/wing/httpd_1.1/cgi-src/

此外,要修改 /etc/service 檔,將 80 port 開給 HTTPD 使用

#

Network services, Internet style

#

```

#    @(#)services 5.8 (Berkeley) 5/9/91
#
echo      7/tcp
echo      7/udp
discard   9/tcp      sink null
discard   9/udp      sink null
systat    11/tcp      users
daytime   13/tcp
daytime   13/udp
netstat   15/tcp
qotd      17/tcp      quote
chargen   19/tcp      ttytst source
chargen   19/udp      ttytst source
ftp       21/tcp
telnet    23/tcp
smtp      25/tcp      mail
time      37/tcp      timserver
time      37/udp      timserver
rtp       39/udp      resource    # resource location
nameserver 42/tcp      name        # IEN 116
whois     43/tcp      nickname
domain    53/tcp      nameserver  # name-domain server
domain    53/udp      nameserver
mtp       57/tcp
# Bootp experimental (sellgren@vangogh)
bootp     67/udp      # bootp server
#bootpc   68/udp      # bootp client
#

```

tftp	69/udp		
rje	77/tcp	netrjs	
finger	79/tcp		
chimera	80/tcp		
link	87/tcp	ttylink	
supdup	95/tcp		
hostnames	101/tcp	hostname	# usually from sri-nic
tsap	102/tcp		# part of ISODE.
#csnet-cs	105/?		
pop	109/tcp	postoffice	
sunrpc	111/tcp		
sunrpc	111/udp		
auth	113/tcp	authentication	
sftp	115/tcp		
uucp-path	117/tcp		
nntp	119/tcp	readnews untp	# USENET News Transfer Protocol
ntp	123/udp		# Network Time Protocol
snmp	161/udp		
snmp-trap	162/udp		
#			
# UNIX specific services			
#			
exec	512/tcp		
biff	512/udp	comsat	
login	513/tcp		
who	513/udp	whod	
shell	514/tcp	cmd	# no passwords used
syslog	514/udp		

```

printer    515/tcp    spooler    # line printer spooler
talk       517/udp
ntalk      518/udp
route      520/udp    router routed
timed      525/udp    timeserver
tempo      526/tcp    newdate
courier    530/tcp    rpc
conference 531/tcp    chat
netnews    532/tcp    readnews
netwall    533/udp                    # -for emergency broadcasts
uucp       540/tcp    uucpd      # uucp daemon
remotefs   556/tcp    rfs_server rfs # Brunhoff remote filesystem
ingreslock 1524/tcp
#
# Kerberos (Project Athena/MIT) services
#
kerberos   750/udp    kdc        # Kerberos (server) udp
kerberos   750/tcp    kdc        # Kerberos (server) tcp
krbupdate  760/tcp    kreg       # Kerberos registration
kpasswd    761/tcp    kpwd       # Kerberos "passwd"
klogin     543/tcp                    # Kerberos rlogin
eklogin    2105/tcp                   # Kerberos encrypted rlogin
kshell     544/tcp    krcmd      # Kerberos remote shell
#end of service file

```

接著,以 root 執行 httpd ,便成功的完成安裝了。

5.6.6 測試:

我們以 Windows 的 Mosaic, 386BSD 下的 Mosaic 以及 lynx (一種WWW的 command-line client)測試,證實 httpd 已能正常的執行,並且送出中文的hypertext file。

5、7節 中文化X Window之建置

5.7.1 前言：

X Window系統是爲了實現雅典娜計畫(Project Athena)的需求而設計出來的，它係由美國麻省理工學院(M.I.T)與DEC、HP、與IBM等數家大電腦公司所開發的一套視窗軟體。它以分散式的"委託者／伺服器"(Client/Server)模式爲架構，具備網路透通性(Network Transparency)與設備無關性(device independent)等特性。因此，我們希望藉由X Window的圖型介面來設計使用者中文輔助系統。

5.7.2 研究目的：

嘗試將X Window中文化，包括所有訊息、視窗的標題(Window Title)、選單項目、以及視窗內文字。如此可讓使用者很輕鬆地使用與觀看線上輔助畫面。

5.7.3 研究方法：

圖一是我們研究的流程。

5.7.4 實驗部份：

5.7.4.1 安裝CXTerm

將CXTerm的原始碼自 Anonymous FTP 站取回，在樣本機器上重新編譯(Compile)。此外，取回CXTerm所使用之中文字型，將其安裝至 /usr/X386/lib/X11/fonts/chinese 目錄下，並修改Xconfig的內容，以使其成爲XFree86的內定字型。

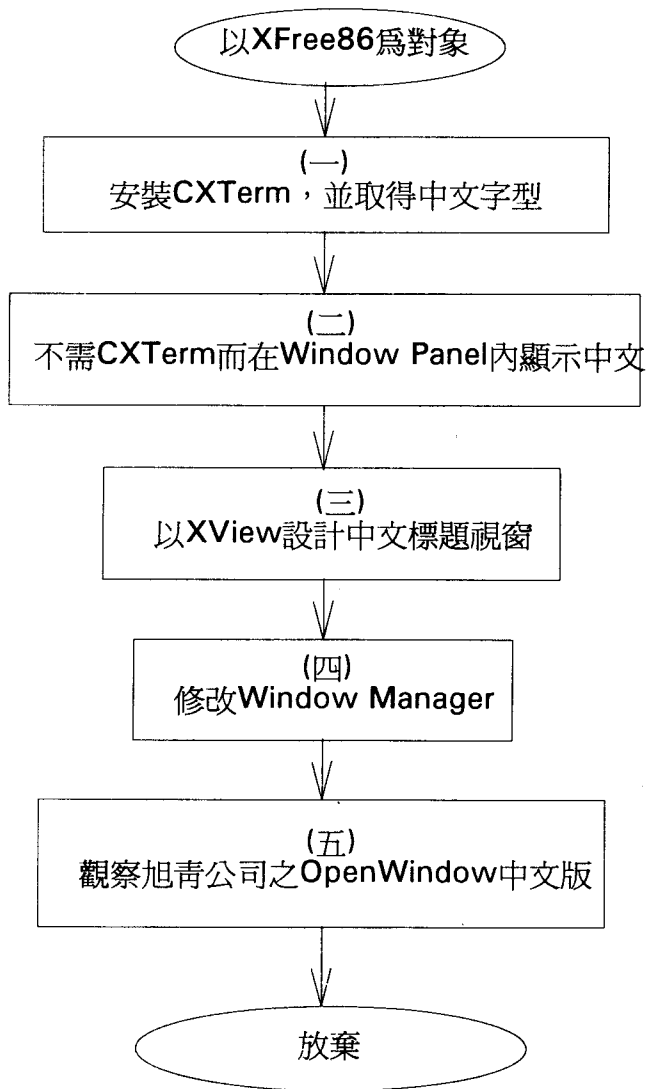
CXterm是目前國內在X Window系統下使用最廣泛之中文介面。它提供顯示與輸入中文，並有多種中文輸入法。此外，CXTerm也是唯一在X Window下非商業軟體的中文介面程式，很容易取得程式原始碼。因此，我們第一步便是觀察CXTerm的中文顯示方式。

觀察的結果發現，CXTerm是先取得中文字型的架構，然後以繪圖方式將中文字顯示於Window Panel內；因此，我們便嘗試以程式在Window Panel內顯示中文。

5.7.4.2 在Window Panel內顯示中文

我們希望在X Window下顯示中文；然而，由於對X Window Programming並不熟悉，所以我們就尋找是否有類似功能的程式範例。在ifcss.org的software/x-win目錄下，有一xhzview.3-0.tar.Z的範例程式。取回後，以`zcat xhzview.3-0.tar.Z | tar xvf` -解開，其主程式xhzview.c的原始碼列表如附錄一所示。

雖然此範例程式可在X Window下將CXTerm所附的中文字型顯示出來，然而也只能在Window Panel內顯示。對於Window標題，它是直接傳一字串給X Server來顯示，並不是自己處理。所以我們希望以程式來指定字型與中文標題，XView便成了我們第一選擇。



圖一 X Window中文化嘗試過程

5.7.4.3 以XView設計中文標題視窗

XView(X Window-System-based Visual/Integrated Environment for Workstation)是一個支援在X Window系統下執行圖型基礎應用程式的使用者介面。透過XView，程式設計師可以很容易地發展X Window的應用程式。因此，我們希望以XView來製作一中文視窗。

附錄二是我們的測試程式。程式中直接給與中文字到Window標題與Window內的按鈕。執行結果顯示，在Window標題與按鈕會出現奇怪的符號，並沒有中文字出現，也不是將8位元中文內碼消除1位元所對應的英文字。

由以上結果可知，8位元的中文內碼可順利傳為參數；然而Window Manager並不能將對應的中文字型取出，因此便以英文字型超過ASCII 128的特殊符號表示。所以接下來我們便針對Window Manager下手。

5.7.4.4 修改Window Manager

我們初步判斷Window Manager可以設定顯示的字型，所以我便使用fvwm與olwm兩種不同的Window Manager來設定。

首先對預設字型作修改。在fvwm方面，將fvwmrc中的數行設定為

```
Font et-ch16
```

```
WindowFont et-ch16
```

```
IconFont et-ch16
```

使其顯示中文字型；而在olwm方面，以

```
olwm -font et-ch16
```

來設定其顯示字型為中文字型。結果發現，Window Title根本無法顯示任何字，而在按鈕上仍然是一樣的亂碼出現。

我們懷疑是Window Manager的原始程式中有超過ASCII碼128便不顯示的定義，於是便搜尋原始碼中的7f, 127, 或128的字串，將其修改為ff或255，再重新編譯。結果仍然是一樣的情況。可見在X Window下顯示中文字並不只是Window Manager的部份便可辦到。

5.7.4.5 觀察旭青公司之Open Window中文版

在不得其門而入的情形下，我們得知旭青公司有一套Open Window的中文版，並且在中央大學管理學院電算中心有購置一套。因此我們就想觀察其如何將X Window中文化。

觀察結果發現，旭青公司的X Window中文版所自修改原英文版修改的部份並不只是Window Manager而已。包含低階的X Libraries與其所附之應用程式都必須一起更動。如此浩大的工程是動用不少的人力與時間，決非我們在少數組員及有限時間的情況下能完成。因此，我們很沉痛地放棄了X Window中文化的構想。希望未來有更多有志者能夠依此經驗完成這個願望，以造福無法購買昂貴商業軟體的中文軟體愛好者。

5.7.5 結果：

雖然我們沒有完成所預定的目標，但是整個試驗過程當中，也有不少發現。例如CXTerm的中文字型存圖方式與英文字型有很大差異，因此原來X Window的顯示方式便不能順利地以指定字型來顯示中文字型。

5.7.6 討論：

此次不能完成X Window中文化，除了人力與時間上缺乏的因素外，我們對於X Window的陌生也是重要因素。由於並不十分熟悉X Window的成式設計，使得我們無法以低階的工具，如X Libraries，來設計中文介面。在參考資料的收集方面也做得並不很齊全，使得有些現象無法理解。例如以xlsfonts -l所得到的中文字型與英文字型在某些欄位有很大的差異，手上卻沒有詳細解釋各欄位用途的資料。而憑著在news或BBS上的詢問皆不能得到滿意的回答。

總之，誠心希望有對X Window熟悉的前輩能給予我們意見與指導，使得有朝一日，在FTP站都能拿到以自己國家文字顯示的X Window。

5.7.7 附錄：

5.7.7.1 附錄一：xhzview.c列表

```
/*
```

```
xhzview.c
```

```
Copyright (C) 1989,1990,1991  Ricky Yeung
```

```
A simple program to display mixed 8-bit or 16-bit characters  
on X window. It is intended for viewing Chinese code files  
(e.g. GB or BIG5).
```

```
History:
```

```
06/03/91  No need to use wsimple, use Xt instead  
          Fixed a "bug" in line end calculation  
          Lots of clean up  
08/26/90  Version 2.0  
          Eliminate the -f option  
          Eliminate tempfile if not read from stdin  
          More browsing control  
          Use line ends instead of sequential reading  
          Allow resize of window  
          Help facility
```

Did some cleanup
10/05/89 No longer requires cclwrap filter.
Can read directly from stdin.
Some minor bugs fixed.
09/26/89 First working version.

Xhzview is free software and everyone is permitted to copy and distribute verbatim copies of this document, but changing it is not allowed.

This software is provided "as is" and the author disclaim all warranties with regard to this software.

This software has no connection with my employer.

Comments, bug reports, or suggestions should be sent to
yeung@decwrl.dec.com or yeung@atd.dec.com

*/

```
#include <stdio.h>
#include <ctype.h>
#include <X11/Intrinsic.h>
#include <X11/StringDefs.h>
#include <X11/Xlib.h>
#include <X11/Shell.h>

#define XHZVIEW_VERSION "xhzview 3.0"

/* default parameters */
#ifndef DEFAULT_HANZI_FONT
#define DEFAULT_HANZI_FONT "beijing24"
#endif
#ifndef DEFAULT_ASCII_FONT
#define DEFAULT_ASCII_FONT "vg-20"
#endif
#ifndef DEFAULT_LINE_SPACING
#define DEFAULT_LINE_SPACING "4"
#endif
#ifndef DEFAULT_CHAR_SPACING
#define DEFAULT_CHAR_SPACING "0"
#endif
#ifndef DEFAULT_LINES
#define DEFAULT_LINES "24"
#endif
#ifndef DEFAULT_CHARS
#define DEFAULT_CHARS "40"
```

```

#endif
#ifndef DEFAULT_LINE_WRAP
#define DEFAULT_LINE_WRAP    "FALSE"
#endif

#define LINELEN 3000 /* size of buffer */

#define  NLINES 100 /* amount to realloc for line ends */

#define  MIN(a,b) (((a)<(b)) ? (a) : (b))
#define  MAX(a,b)  (((a)>(b)) ? (a) : (b))
#define  ISHZ(c)   ((c) & 0x80)

extern char *malloc(), realloc();

typedef struct
{
    char *c_font;      /* Chinese Font name */
    char *e_font;      /* English Font name */
    int lnSpacing;     /* default line spacing */
    int chSpacing;     /* space between ascii & hanzi */
    int lines;         /* default lines / page */
    int chars;         /* default chars / line */
    int noLnWrap;      /* run through line wrap filter? */
} mydata, *mydataptr;

char *filename = NULL;      /* input file name, empty for stdin */
char tempfile[LINELEN]="/usr/tmp/xhzview0_XXXXXX"; /* tmp file */

int curLn = 0;              /* current line number */
int cLnHt;                  /* line height */

char line[LINELEN];        /* line buffer */

FILE *fp = NULL;

static  int  lp;            /* # line-ends */
static  int  nLn;          /* number of total line-ends allocated */
static  struct lnType      /* line-end structure */
{
    int st;                /* start position */
    long n;                /* number of characters */
} *lns = NULL;

mydata data;

```

```

Display *dpy;
Window wind;          /* drawing toplevel window */
GC gc;                /* default GC */
char *geometry;

XFontStruct *c_font;  /* Chinese font */
XFontStruct *e_font;  /* English font */

/* Toolkit resources */
static XtResource resources[] = {
    { XtNfont, XtCFont, XtRString, sizeof(char *),
      XtOffset(mydataptr, e_font), XtRString, DEFAULT_ASCII_FONT},
    { "cFont", "CFont", XtRString, sizeof(char *),
      XtOffset(mydataptr, c_font), XtRString, DEFAULT_HANZI_FONT},
    { "lnSpacing", "LnSpacing", XtRInt, sizeof(int),
      XtOffset(mydataptr, lnSpacing), XtRString, DEFAULT_LINE_SPACING},
    { "chSpacing", "ChSpacing", XtRInt, sizeof(int),
      XtOffset(mydataptr, chSpacing), XtRString, DEFAULT_CHAR_SPACING},
    { "lines", "Lines", XtRInt, sizeof(int),
      XtOffset(mydataptr, lines), XtRString, DEFAULT_LINES},
    { "chars", "Chars", XtRInt, sizeof(int),
      XtOffset(mydataptr, chars), XtRString, DEFAULT_CHARS},
    { "noLnWrap", "NoLnWrap", XtRBoolean, sizeof(Boolean),
      XtOffset(mydataptr, noLnWrap), XtRString, DEFAULT_LINE_WRAP},
};

/* command line options for resources */
static XrmOptionDescRec options[] = {
    {"-cFont", "*cFont", XrmoptionSepArg, NULL},
    {"-cfn", "*cFont", XrmoptionSepArg, NULL},
    {"-afn", "*Font", XrmoptionSepArg, NULL}, /* for backward compatibility*/
    {"-ls", "*lnSpacing", XrmoptionSepArg, NULL},
    {"-cs", "*chSpacing", XrmoptionSepArg, NULL},
    {"-l", "*lines", XrmoptionSepArg, NULL},
    {"-n", "*chars", XrmoptionSepArg, NULL},
    {"-nowrap", "*noLnWrap", XrmoptionNoArg, "TRUE"},
};

/*
 *-----
 * Help routines
 *-----
 */
usage(program_name)
    char *program_name;
{

```

```

    fprintf(stderr, "usage: %s [-cfn <HanziFont>] [-fn <AsciiFont>] [-nowrap] \\ \n",
program_name);
    fprintf(stderr, "      [-ls <ls>] [-cs <cs>] [-l <l>] [-n <n>] [filename]\n");
    fprintf(stderr, " <HanziFont> - Default is '%s'.\n", DEFAULT_HANZI_FONT);
    fprintf(stderr, " <AsciiFont> - Default is '%s'.\n", DEFAULT_ASCII_FONT);
    fprintf(stderr, " <ls> - space between lines, 1/<ls> of the max. line height, \n");
    fprintf(stderr, "      0 for none. Default is %s.\n",
        DEFAULT_LINE_SPACING);
    fprintf(stderr, " <cs> - space between Ascii/hanzi, 1/<cs> of hanzi width,\n");
    fprintf(stderr, "      0 for none. Default is %s.\n",
        DEFAULT_CHAR_SPACING);
    fprintf(stderr, " <l> - # lines per page. Default is %s.\n",
        DEFAULT_LINES);
    fprintf(stderr, " <n> - # characters per line. Default is %s.\n",
        DEFAULT_CHARS);
    fprintf(stderr, " The -nowrap option tells not to wrap long lines.\n");
    exit(0);
}

```

help()

```

{
    printf("\n<n> <Left Mouse Button> - go backward n pages (default 1)\n");
    printf("<n> <Right Mouse Button> - go forward n pages (default 1)\n");
    printf("h - this help message\n");
    printf("q - quit\n");
    printf("<- go to the first page of file\n");
    printf("> - go to the last page of file\n");
    printf("<n> f - go forward n pages (default 1)\n");
    printf("<n> b - go backward n pages (default 1)\n");
    printf("<n> <RET> - go forward n lines (default 1)\n");
    printf("<n> p - go backward n lines (default 1)\n");
    printf("<n> g - go to the n-th line (default line 1)\n");
}

```

/*

*-----

* general file I/O and misc routines

*-----

*/

MyExit(i)

int i;

{

/* clean up before exit */

/* remove tempfile unless input is a file */

if (!filename) unlink(tempfile);

exit(i);

```

}

Fatal_Error(s)
    char *s;
    {
        fprintf(stderr, "%s\n", s);
        MyExit(1);
    }

void Beep()
    {
        XBell(dpy, 50);
    }

/* open a file with error check */
FILE *Myfopen (file, mode)
    char *file, *mode;
    {
        FILE *fp;

        if (!(fp=fopen(file, mode)))
            {
                fprintf(stderr, "Can't open file %s.\n", file);
                MyExit(1);
            }
        return (fp);
    }

/*
    Read the i-th line from file fp, using line-ends info from lns.
    The line is stored in the character array line.
    */
readLn(fp, lns, line, i)
    FILE *fp;
    struct lnType *lns;
    char *line;
    int i;
    {
        register    int j;

#ifdef SEEK_SET
#define SEEK_SET 0
#endif

        fseek(fp, lns[i].st, SEEK_SET);
        for (j=0; j<lns[i].n; j++)

```

```

        line[j] = fgetc(fp);
        line[lns[i].n] = '\0';
    }

/* copy srcfile to dstfile, if pointer empty, use stdin/stdout */
copy(dstfile, srcfile)
    char *dstfile, *srcfile;
{
    FILE *ifp, *ofp;
    int c;

    ifp = (srcfile) ? fopen(srcfile, "r") : stdin;
    ofp = (dstfile) ? fopen(dstfile, "w") : stdout;

    while ((c=fgetc(ifp))!=EOF)
        fputc(c, ofp);
    fclose(ifp);
    fclose(ofp);
}

XFontStruct *MyOpenFont(name)
    char *name;
{
    XFontStruct *font;

    if (!(font=XLoadQueryFont(dpy, name)))
    {
        sprintf(line, "Unable to open font %s!", name);
        Fatal_Error(line);
    }

    return(font);
}

/*
 *-----
 * Page Display routines
 *-----
 */
MyDrawString(s, n, ch16, x, y)
    char *s;
    int n, ch16, x, y;
{
    XGCValues tmp_gc;

    if (ch16)

```



```

{
    tmp_gc.font = c_font->fid;
    XChangeGC(dpy, gc, GCFont, &tmp_gc);
    /* two bytes counted as 1 char */
    XDrawImageString16(dpy, wind, gc, x, y, s, n / 2);
    return(x + (n/2) * c_font->max_bounds.width + data.chSpacing);
    /*
     * XTextWidth doesn't count the undefined chars, so can't use
     * XTextWidth16(c_font, s, n)
     */
}
else
{
    tmp_gc.font = e_font->fid;
    XChangeGC(dpy, gc, GCFont, &tmp_gc);
    XDrawImageString(dpy, wind, gc, x, y-e_font->descent, s, n);
    return(x + XTextWidth(e_font, s, n) + data.chSpacing);
}
}

```

```

/* draws a line in the file at vertical position y */
MyDrawLine(y, line)
    int y;
    char *line;
{
    register int i = 0;
    int s = 0, x = 1, len = strlen(line), lastMode, curMode;

    if (len==0) return;

    /* check if high bit set to determine initial mode */
    lastMode = ISHZ(line[0]);

    for (i=0; i<len; ++i)
    {
        curMode = ISHZ(line[i]); /* check if high bit set */

        if (curMode!=lastMode)
        {
            x = MyDrawString(line + s, i - s, lastMode, x, y);
            s = i; /* update start of next run */
            lastMode = curMode; /* remember "previous" mode */
        }

        if (curMode) ++i; /* skip 2nd byte of two-byte char */
    }
}

```

```

    /* take care the left-over */
    MyDrawString(line + s, i - s, curMode, x, y);
}

/*
display a page starting from curLn
handles resized window as well
*/
Display_Page()
{
    int i=0, y, end;
    static int lastLn = -1;

    if (lastLn!=curLn) XClearWindow(dpy, wind);
    lastLn = curLn;
    y = c_font->ascent;
    end = curLn + MIN(data.lines + curLn, lp - curLn);
    for (i=curLn; i<end; i++)
    {
        readLn(fp, lns, line, i);
        MyDrawLine(y, line);
        y += cLnHt;
    }
}

/* calculate the window's minimum size & increment */
Calc_Default_Size(shell)
Widget shell;
{
    Arg  args[10];
    int n = 0;
    int width = c_font->max_bounds.width;

    XtSetArg(args[0], XtNgeometry, &geometry);
    XtGetValues(shell, args, 1);

    if (!geometry)
    {
        /* user didn't override, use ideal size */
        /* leave one extra space at the right edge */
        XtSetArg(args[n], XtNwidth, width * (data.chars + 1));    n++;
        XtSetArg(args[n], XtNheight, cLnHt * data.lines);    n++;
    }

    XtSetArg(args[n], XtNwidthInc, width);    n++;
}

```

```

XtSetArg(args[n], XtNheightInc, cLnHt); n++;
XtSetArg(args[n], XtNminWidth, width); n++;
XtSetArg(args[n], XtNminHeight, cLnHt); n++;
XtSetValues(shell, args, n);
}

/*
 *-----
 * Navigation routines
 *-----
 */
/* go backward n pages */
Go_Back(n)
    int n;
{
    if (n==0) n=1;
    if (curLn==0)
    {
        Beep();
        return;
    }
    else if ((curLn - n * data.lines)<0) curLn=0;
    else curLn -= n * data.lines;
    Display_Page();
}

/* go forward n pages */
Go_Forward(n)
    int n;
{
    if (n==0) n=1;
    if ((curLn + n * data.lines)>=lp)
    {
        Beep();
        return;
    }
    else
    {
        curLn += n * data.lines;
        Display_Page();
    }
}

Forward_Line(n)
    int n;
{

```

```

int i;

i = curLn + n;
if ((i>=lp)||(i<0))
{
    Beep();
    return;
}
curLn += n;
Display_Page();
}

/* go to the last page */
Go_End()
{
    curLn = lp - data.lines;
    if (curLn<0) curLn = 0;
    Display_Page();
}

Do_Key(c)
char c;
{
    static int n = 0;

    if (isdigit(c))
        n = n * 10 + (c - '0');
    else
    {
        switch (c)
        {
            case 'q':
                MyExit(0);
                break;
            case ' ':
            case 'f':
                Go_Forward(n);
                break;
            case 'b':
                Go_Back(n);
                break;
            case '>':
                Go_End();
                break;
            case '<':
                curLn=0;

```

```

    Display_Page();
    break;
case 'h':
    help();
    break;
case '\015':      /* RET */
    Forward_Line((n)?n:1);
    break;
case 'p':
    Forward_Line((n)?-n:-1);
    break;
case 'g':
    if (n==0) n = 1;
    else if (n>=lp)
    {
        Beep();
        break;
    }
    curLn = n - 1;
    Display_Page();
    break;
default:
    Beep();
    break;
}
n = 0;
}
}

/*
*-----
* line end calculation routines
*-----
*/
/* set the line end of current line and line start of next line */
SetLnEnd(n, i)
int n, i;
{
    lns[lp++].n = n;

    if (lp>=nLn)
    {
        /* if no more line-ends, allocate more */
        nLn += NLINES;
        lns = (struct lnType *)
            realloc(lns, nLn * sizeof(struct lnType));
    }
}

```

```

        if (!lns)
            Fatal_Error("Not enough memory.\n");
    }

    lns[lp].st = i;
}

/* allocate line-end structure */
ResetLnEnds()
{
    if (lns) free(lns);    /* reset lns data */
    nLn = NLINES;
    lns = (struct LnType *) malloc(NLINES * sizeof(struct LnType));
    lp = 0;
    if (!lns) Fatal_Error("Not enough memory.\n");
}

/* calculate line-ends */
Cal_LnEnds(wrap)
    int wrap;
{
    int i = 0;    /* character count */
    int n = 0;    /* char count for current line */
    int lnChrs;    /* chars per line */
    int c, ishz = 0;
    FILE *ifp;

    ResetLnEnds();

    lns[0].st = 0;
    ifp = Myfopen(tempfile, "r");

    lnChrs = data.chars * 2;

    while ((c=fgetc(ifp))!=EOF)
    {
        n++;
        i++;

        if ((n>=lnChrs)&&wrap)    /* wrap a line */
        {
            /* if this one is 1st byte of hanzi, don't count it at same line */
            if ((!ishz)&&(ISHZ(c)))
            {
                SetLnEnd(n-1, i-1);
            }
        }
    }
}

```

```

        n = ishz = 1;
    }
    else {
        SetLnEnd(n, i);
        n = ishz = 0;
    }
    continue;
}
if (c=='\n')
{
    /* to skip the '\n' character, SetLnEnd(n-1, ++i); */
    SetLnEnd(n, i);
    n = ishz = 0;
    continue;
}
if (ishz) ishz = !ishz;    /* 2nd byte, reset ishz flag */
else ishz = ISHZ(c);    /* see if 1st byte */
}
fclose(ifp);
}

/*
 *-----
 * interaction handlers
 *-----
 */
expose_handler(w, data, event)
Widget      w;
caddr_t     data;
XEvent      *event;
{
    if (event->xexpose.count==0)
        Display_Page();
}

buttonPress_handler(w, data, event)
Widget      w;
caddr_t     data;
XEvent      *event;
{
    if (event->xbutton.button==1) Do_Key('b');
    if (event->xbutton.button==2) MyExit(0);
    if (event->xbutton.button==3) Do_Key('f');
}

keyPress_handler(w, data, event)

```

```

Widget    w;
caddr_t   data;
XEvent    *event;
{
    register int i;
    char buffer[10];

    i = XLookupString(event, buffer, 10, 0, 0);
    if (i==1) Do_Key(buffer[0]);
}

resize_handler(w, cdata, event)
    Widget    w;
    caddr_t   cdata;
    XEvent    *event;
{
    Arg args[2];
    int olp = lp, n = 0;
    Dimension ht, width;
    static Dimension lastHt = 0;
    static Dimension lastWidth = 0;

    XtSetArg(args[n], XtNwidth, &width);    n++;
    XtSetArg(args[n], XtNheight, &ht);      n++;
    XtGetValues(w, args, n);
    if ((lastHt!=ht)||((lastWidth!=width))
        {
            data.lines = ht / cLnHt;
            data.chars = width / c_font->max_bounds.width;
            Cal_LnEnds(!data.noLnWrap);
            curLn = curLn * lp / olp;        /* locate appro. same line */
            XClearWindow(dpy, wind);
            Display_Page();
            lastHt = ht;
            lastWidth = width;
        }
    }

/*
 *-----
 * main program
 *-----
 */
main(argc, argv)
    int    argc;
    char  **argv;

```



```

{
Widget toplevel;
XGCValues gc_init;
Arg args[10];
int n = 0;

/* initialize toolkit and process command line options */
toplevel = XtInitialize("xhzview", "Xhzview", options, XtNumber(options),
                      &argc, argv);

if (argc>2) usage(argv[0]);
if (argc==2) filename = argv[1];

/* process resources */
XtGetApplicationResources(toplevel, &data, resources,
                          XtNumber(resources), NULL, 0);

dpy = XtDisplay(toplevel);
c_font = MyOpenFont(data.c_font);
e_font = MyOpenFont(data.e_font);

/* calculate line height, spacing, etc. */
cLnHt = MAX(c_font->ascent + c_font->descent,
            e_font->ascent + e_font->descent);
if (data.lnSpacing) data.lnSpacing = cLnHt / data.lnSpacing;
if (data.chSpacing)
    data.chSpacing = c_font->max_bounds.width / data.chSpacing;
cLnHt += data.lnSpacing;

/* Get minimum preferred size */
Calc_Default_Size(toplevel);

XtRealizeWidget(toplevel);

wind = XtWindow(toplevel);

if (filename)
    sprintf(line, "%s -- %s", XHZVIEW_VERSION, filename);
else sprintf(line, "%s", XHZVIEW_VERSION);

XStoreName(dpy, wind, line);
XSetIconName(dpy, wind, line);

/* obtain GC */
gc_init.font = c_font->fid;
XtSetArg(args[n], XtNbackground, &gc_init.background);    n++;

```

```

XtSetArg(args[n], XtNforeground, &gc_init.foreground);    n++;
XtGetValues(toplevel, args, n);
gc = XtGetGC(toplevel, GCFont|GCForeground|GCBackground, &gc_init);

mktemp(tempfile);

if (!filename) copy(tempfile, NULL);
else strcpy(tempfile, filename);

fp = Myfopen(tempfile, "r");

Cal_LnEnds(!data.noLnWrap);

XtAddEventHandler(toplevel, ExposureMask, False, expose_handler, NULL);
XtAddEventHandler(toplevel, ButtonPressMask, False,
    buttonPress_handler, NULL);
XtAddEventHandler(toplevel, KeyPressMask, False, keyPress_handler, NULL);
XtAddEventHandler(toplevel, StructureNotifyMask, False, resize_handler, NULL);

XtMainLoop();
}

```

5.7.7.2 附錄二：XView測試程式

```
/*
 * quit.c -- test program to display Chinese on window title and
 *          panel button.
 */

#include <xview/xview.h>
#include <xview/frame.h>
#include <xview/panel.h>

Frame frame;

main(argc, argv)
int argc;
char *argv[];
{
    Panel panel;
    void quit();

    xv_init(XV_INIT_ARGC_PTR_ARGV, &argc, argv, NULL);

    frame = (Frame)xv_create(NULL, FRAME,
        FRAME_LABEL, "測試程式",
        XV_WIDTH, 200,
        XV_HEIGHT, 100,
        NULL);

    panel = (Panel)xv_create(frame, PANEL, NULL);

    (void)xv_create(panel, PANEL_BUTTON,
        PANEL_LABEL_STRING, "離開",
        PANEL_NOTIFY_PROC, quit,
        NULL);

    xv_main_loop(frame);
    exit(0);
}

void
quit()
{
    xv_destroy_safe(frame);
}
```

附錄 5 A 利用Hytelnet建置簡易使用者中文輔助系統之方法

A.1 前言:

在 5.5 hytelnet 的建置中我們已經對 hytelnet 的功能及工作原理做了詳細的說明,我們知道 hytelnet 最好用的功能乃是做檔案的分類及讀取,加上簡易好學的操作。這些都是一個好的線上輔助系統所必需具備的。所以我們選用 hytelnet 作為我們線上輔助系統的藍本,加以修改成為一個提供使用者查詢的程式。

A.2 系統建置:

我們在前文已詳述了如何將 hytelnet 中文化,而對其工作原理請參閱 5.5 的圖一,我們是跟據此一原理建置我們的線上輔助系統的。

首先,我們要自行編寫一個 start 檔 (即執行 hytelnet 後所看到的主選單): 並將所欲提供查詢資料檔案和 hytelnet 此執行檔一起放入hytelnet 此目錄中。(放在同一目錄中)

修改完後再重新 make 一次 即可:

make hytelnet.c

執行 hytelnet 即出現主選單 (start 檔), 供使用者選擇如(圖一)。

A.3 使用方式 :

(1) 在 PC 上機則必需先載入倚天, 若是在 X-WINDOWS 則開啓 CXTerm。

(2) 然後執行 hytelnet 則進入輔助系統的主畫面, 即前面的 start 檔。

* 歡迎利用線上輔助系統 *

* Jan 20, 1994 *

安裝 FreeBSD 查詢	<Bsd>
安裝 X-WINDOWS 查詢	<Xwin>
安裝 Cxterm 查詢	<Cxterm>
安裝 X-View 查詢	<Xview>
安裝 GhostScript & GhostView查詢	<Ghost>

m 返回主畫面 q 離開線上輔助系統

.....

h 查詢系統自我簡介！

.....

上鍵 前一頁 左鍵 前一層選項
下鍵 下一頁 右鍵 下一層選項

(圖一) start

操作上很容易，只要上下左右鍵即可：

關於操作的說明在操作中按下 'h' 鍵亦可看到文件說明，按 m 則不管在第幾層選單，直接跳回第一層選單(start 檔)。

例如：選擇 --> 安裝 GhostScript & GhostView查詢 <Ghost> 在進入
hytelnet 程式後,你會發現有一個字串是呈現反白的狀態此時,可以用下鍵將之移動
到不同的字串之上。

a. 將反白游標移到 <Ghost> 處。按 enter ,出現以下畫面

```
*****
| 安裝介紹 Ghostview-2.6.1 & Ghostview-1.5      |
*****
Ghostscript-2.6.1 相關資料 <Gho0>
Ghostview-1.5 相關資料 <Gho1>
```

m 返回主畫面 q 離開線上輔助系統

.....

h for help anytime !

.....

上鍵 前一頁 左鍵 前一層選項
下鍵 下一頁 右鍵 下一層選項

b. 選擇 `ghostview-1.5`，將游標移到 `<Ghv>`，按 `enter` 則出現 `ghostview-1.5` 相關資料。在以上下鍵檢視文件上鍵表示前一頁，下鍵表示下一頁。

c. 以左右鍵切換選單，左鍵表示前一層選單，右鍵表示下一層選單。

按左鍵回到 (b) 之畫面。

再按左鍵回到 (a) 之畫面。

d. 按 `h` 則出現線上輔助系統的自我簡介。

e. 按 `q` 則跳出線上輔助系統。

而這些功能鍵都可以自行在 `hytelnet.c` 中訂定。

附錄 5 B 利用Http建置簡易使用者中文輔助系統之方法

B.1 建立一個WWW server：

B.1.1 為何使用WWW來做中文輔助系統：

使用 `hytelnet` 既然已經可以做為線上輔助系統，為何我們又要使用WWW來做線上的輔助呢？因為WWW可以提供更多 `hytelnet` 所沒有的功能，例如：圖形的解說及語音等多媒體的服務。前面說過，WWW所提供的是超媒體的服務，包括了動畫、圖形、語音、文字等等，而 `hytelnet` 所能提供的服務卻僅限於文字。因此，我們希望能以WWW做更多的線上服務。

B.1.2 如何以WWW做為中文的線上輔助？

利用WWW建置一個輔助系統，首先就是把Server建立起來，我們選用的server版本是HTTPD1.1(建立方式參見5.6)，然後剩下的工作就是HTML文件的編寫。使用者可以透過一些所謂的瀏覽工具如：`Mosaic`，`MidasWWW`，`Lynx`(這是我們目前所

知可以讀取中文文件的Client)來讀取我們放在 Server上的HTMP文件。在附錄C.2中我們將介紹如何編寫HTML文件來建立這使用者中文輔助系統。此時,我們若要對使用者提供線上服務,我們只要把系統上的 WWW-client 的內定 server 指向自己的 server 端,使用者便可以讀到我們想提供的訊息。當然,遠端的使用者若想讀到我們提供的訊息,也可以指定到我們的 server 上來讀取資料,只要在執行 WWW-client 時多加一行如下的參數:

http://<machine-name>/

例如:

http://athena2.net.ncu.edu.tw/ 或 http://140.115.10.121/

便可讀到我們所提供的資料了。

B.2: HTML 檔的說明(由蕭松瀛同學提供)

Html 是一種超文件 (HyperTEXT) 檔案,經由一些特殊的工具你可以看到超文件的內容。它並沒有什麼特別的地方,因此你可以用一般的編輯器來編輯你需要的檔案(文字檔),這些檔案經過一些處理後(加上一些符號及字母)就變成所謂的超文件。當然它是和一般的文字檔有所區別的,超文件可以被放在一些 Server (如 Word Wide Web Server)上,再由其它的工具(如 Cello, Mosaic, lynx..)來閱讀。超文件和一般的文字檔最大的不同在於他經由工具秀出來後可以做許多的變化,如文字的大小、顏色的變化,或是指向其他的檔案、圖形、甚至聲音等等。超文件迷人的地方就在這裡。

而 Windows 下的 HyperEdit (你可以在 ftp.cam.org 裡 /systems/ms-windows/slip-ppp/SPECIAL 下找到 htmledit.zip) 就是一個編輯超文件的工具。

超文件中語法的格式都是以 <form> 開頭,而以</form> 結束,例外的只有換行符號 <P>,它不需要 </P> 來表示結束。當你的工具讀到 <form> 時,他會知道接下來的文字要做特殊的此處理,直到它找到一個相對應的 </form> 時,它才會結束以form 的處理方式(form)只是一個代號而已,並非真正文件中的特殊字)。

同時超文件中的 form 也不分大小寫, 它接受大小寫不同的字。下面我們就來瞧瞧超文件中的符號。

第一：TITLE 標題

`<TITLE> HTML 的簡介 </TITLE>`

基本上每一份文件都有它的一個標題 (title), 它可以用來表示這份文件的主題, 而它應放在文章的最上方。TITLE 在每一種工具所顯示出來的情形都不一樣, 如在Mosaic 中, 它會被放在選單的旁邊, 並不和文章放在一起, 這就依你的工具而不同了。

第二：Headings

`<Hn> HTML 的標題 </Hn> n=1 - 6`

Heading 是文章中的大標題部份, 被他所標記住的部份字體會變黑變粗且放大, 超文章中共提供了六種的 Headings, 通常我們是依照一至六的順序在文章中使用, 也就是先有大標題再有小標題。它常常代表了文章中段落的重點, 例如

`<H1> 第一級的 Heading </H1>`

“第一級的 Heading” 這幾個字將會變成第一級的標題, 秀出來會怎樣, 就要依你所使用的工具設定而定了。

第三：Paragraphs 段落

在超文章中, 它並不承認一般分段的空白或字元, 也就是說文章會一直連下去直到結束為止, 或者是當你的銀幕不夠大時, 它會自動換行而已, 但是不會分段。假如你要它分段, 你必需加上一個特別的分段符號 P。它的特別之處是它沒有回復的</P>。也就是說它只需要 <P>即可。範例如下:

`<TITLE> HTML 的簡介 </TITLE><H1> 第一級的 Heading </H1><P>`

`這是一個段落<P>`

第四：Link 鏈結其它的超文件

超文件中一項很具威力的功能就是它可以呼叫其它的超文件檔案（但必需在你目前所在目錄下的其它檔），也就是說它可以經由你的工具再去 link 其他的資料。而這些被標記的字元會呈現較深的顏色或有底線也就是當你用滑鼠時，它會告訴你它指向那一個超文件。它的標記動作是以 A 為其符號，範例如下：

```
<A HREF ="DEFAULT.HTM">FILE</A>
```

在這行中，它指向的是在你目前目錄下的 DEFAULT.HTM 檔案，而你在工具所見到的卻是呈別種顏色或加上底線的 FILE 你只要將滑鼠移到 FILE 之上，按下它後就或秀出 DEFAULT.HTM 的檔。

第五：URL (Uniform Resource Locator) 格式

超文件中除了上述的連結原檔案所在的目錄下的其他超文件外，它甚至可以到其他的節點去取得其他的超文件檔，或者做出連線的動作。如連接 Gopher，NewServer，ftp server，或再上其它的 WWW Server，WAIS Server 等等。當然它也可以作 telnet 的動作。它的格式如下：

```
<A HREF="scheme://節點：prot/目錄/檔名">超文件中可見到的文字</A>
```

這和上面的例子有點類似，但是它加上了節點(Hostname 或 IP Address)

、port 數字及目錄。而在 scheme 的位置上你可以用下面的幾項選擇：

file：當你要連接自己的系統或別人的 Anonymous FTP Server 時，你必須用 file。

http：你所要連接的一部 WWW Server 時，你需要用此選項。

Gopher：你所連結的檔案在 Gopher Server 時。

WAIS：你要連接的檔案在 WAIS Server 時。

當然你也可以用 telnet 或 News 來作為你的 scheme 只是用到這兩者的機會不大。下面我們來看一個例子：

```
<A HREF ="http://ss.net.ncu.edu.tw/misc/webser/shiao.htm"> shiao's file</A>
```

在這個例子中, `ss.net.ncu.edu.tw` 是一部 `www server` 的節點，而我們要的檔是 `shiao.htm`，它是被放在我們連進去後開始目錄下的 `/misc/webser`，後面的 `shiao's file` 是你用你的工具來看超文件時所可以看到的部份，其他的你都看不到, 所以你只需要將你的滑鼠放在變顏色的 `shiao's file` 上，再壓下，那他就會自動去連啦。

六：Anchor 段落連接

超文件中也可以在本文中連接其他超文件的一個段落，假設你要從原來的文件連 A 文件中的某一段，這時候你就可以用這種方法。首先你必須在 A 文件中建立一個段落(name), 假設我們取名為 `section`，那我們必須在 A 文件中有以下的語法。

```
<A NAME="section">你段落的文章</A>
```

此時，"你段落內的文章"將被標記為 `section`, 而你在本文中要呼叫 `section` 時，你的語法如下：（假設 A 文章的檔案叫 `doc-A.htm`）

```
<A HREF="doc-A.htm#section">link</A>
```

此時，`link` 會呈別種顏色，而你只要壓下 `link` 那他就會自動去找到 A 文章中的 `section` 那一段同時並將他讀出來。此時你要注意的

是你用倒退鍵也無法讀到 A 文件中的其它部份。

七：Lists 排列

超文件提供了下列三種的排列方法，第一種是 **unnumbered**，其次是 **numbered**，最後一種是 **descriptive**。不論何種的排列，在你要排列的主題中最好不要加上分段的符號。首先我們先來瞧瞧 **Unnumbered list**，他的格式如下：

Unnumbered List:

```
<UL>
<LI> 土撥鼠伺服器
<LI> 關於 FTP
<LI> 教育部 Archie 伺服器
</UL>
```

它的輸出結果會在前面加上一個大黑點。或是一個特別的符號。如下(要注意的是 **** 並沒有回復符號，如同 **<P>** 一樣。而 **UL** 表示排列的種類，**LI** 則表示排列的項目 **List Item**。

- 土撥鼠伺服器
- 關於 FTP
- 教育部 Archie 伺服器

Number List:

簡單的說就是有數字的排列，其語法和上述類似，但是將 **UL** 改成 **OL**，底下是它的例子及輸出結果：

```
<OL>
<LI> 土撥鼠伺服器
```

```
<LI> 關於 FTP
<LI> 教育部 Archie 伺服器
</OL>
```

其結果如下：

- 1 土撥鼠伺服器
- 2 關於 FTP
- 3 教育部 Archie 伺服器

Description Lists

這是一種本文排列的語法，它可以讓文章的開頭有空格，同時他也可以讓文章更加有可讀性。更有層次性，我們來看看下面的例子：

```
<DL>
<DT> 這是一篇有關 HTML 的文章
<DD> HTML 是一種超文件的文章，我們可亦藉著一些工具來將他讀出來，這些工具包括了 Mosaic, Cello 等等。</DL>
```

它的輸出將在 <DT> 的那一行移至文區的最前端，而將 <DD> 以下的文章以空兩格的方式排列，結果如下：

```
這是一篇有關 HTML 的文章
    HTML 是一種超文件的文章，我們可亦藉著一些工具來將他讀出來，這些工具包括了 Mosaic, Cello 等等。
```

Nested List

這是一種有層次性的排列，也可以說將上面的幾種語法混和使用，它可以自己包含自己，或是包含別人，下面我們來瞧瞧這個簡單的例子：

```
<UL>
<LI> A artifle of HTML:
  <UL>
    <LI> Title
    <LI> Heading
  </UL>
<LI> List menu
  <OL>
    <LI> Unnumber List
    <LI> Number List
    <LI> Description Lists
  </OL>
</UL>
```

這個短例子的輸出結果如下：

- A artifle of HTML:
 - Title
 - Heading
- List menu
 - 1 Unnumber List
 - 2 Number List
 - 3 Description Lists

八： Preformatted Text

使用 `Pre` 的標籤將會讓你的文章中允許空格，也就是說被 `<PRE>` 及 `</PRE>` 所標記住的範圍，你的文章的原本面貌將會呈現在你所使用的介面中，就如同一般編輯器所編出來的檔案一樣，你將不需要以 `<P>` 來代替換行，而可以直接的秀出你的文章架構，我們來瞧瞧下面的例子：

```
<PRE>
A artifle of HTML:
  Title
  Heading
List menu
  1 Unnumber List
  2 Number List
  3 Description Lists </PRE>
```

則其輸出結果就如同原本的內容，如下

```
A artifle of HTML:
  Title
  Heading
List menu
  1 Unnumber List
  2 Number List
  3 Description Lists
```

九： Blockquote

如果我們將某一部份的文章用 `<blockquote>` 及 `</blockquote>` 標記起來，那

這些被標記的文章在你的工具中秀出來的情形將會變粗變黑，在此我們將不做例子的解說不過它的字型會像放大一樣。

十：Address 位置

在文章中，如果你想要強調你的位置或 E-mail 的地址，此時你可以將你所要強調的東西用<address>及 </address> 標記住，被它所標記住的範圍將會呈現斜體字。

十一：Character formatting 字元的形式

在你的超文件中，文章內的字元可以以好幾種的方式呈現出來，當然他也許依你所使用的工具而有所不同，也就是說你可以在你所使用的工具中定義，譬如<cite>也許被你定義成斜體字，但在別人，卻可能將它定義成別種，這就依人而定了。我們來瞧瞧超文件中所提供的一些定義：

斜體字 (Italic)：

- <I> text </I> 將整篇的文件換成斜體字。
- text
- <cite> text </cite> 當我們引用書名、段落或手冊時使用。
- <var> text </var> 表示一個固定的東西（如 檔名）
- <dfn> text </dfn> 此項功能在 Mosaic 中無效。

粗體字 (Bold):

- text 將文章的字變成粗體。
- test 這也是強調文章。

固定的寬度字形 (Fixed width font)

- <tt> text </tt>
- <code> text </code>

`<kbd> text </kbd>`

十二：特別的字元

在超文件中有三種的字元不能直接秀出來，它必須以特別的方法才能使這些符號在你的工具中展示出來。這三種字元分別是 `<` `>` `&`，要在超文件中出現這三種字元，必須使用跳脫的方式，也就是以其他的碼來解釋它們，我們來看看這三種字元在超文件中的碼：

`<` `<`;
`>` `>`;
`&` `&`;

我們看看下面的句子在超文件中所展示出來的情形，例如：

`<` 是小於的意思，而 `>` 是大於的意思，至於 `&` 則是英文中 `and` 的意思。

上面這段句子將被翻譯成：

`<` 是小於的意思，而 `>` 是大於的意思，至於 `&` 則是英文中 `and` 的意思。

十三：線上圖片 **Inline Image**

NCSA Mosaic 可以在超文件中秀出 **Bitmap** 或是 **GIF** 形式的圖片。每一份圖片都需要花一點時間去傳回來，再展示在你的文件當中。然而需要注意的是目前能展示圖形的超文件工具並不多，而 **Mosaic** 是第一個可以將圖形直接秀出在你的超文件的工具，假如你用別的工具，那你可能會有一些問題。要 **Load** 一份圖片到你的超文件中你必須用下列的語法：

```
<IMG SRC="filename.GIF">
```

此時你的 GIF 檔必須在你目前所在的目錄下。否則你若想到其它的節點去讀圖形檔，你也可以用下面的方法：

```
<IMG SRC="http://hostname/directory/filename.GIF">
```

因為游標的 default 位置, 你的圖形會顯示在原來的那一個段落。

假如你要讀者看到的是圖片和文章接連在一起，而圖片在上方，那你可以加上 ALIGN=top 的命令, 例如：

```
<IMG ALIGN=top SRC="filename.GIF">
```

假如你的圖形很大（也許將近一個螢幕）那你應該在你 load 圖形之前加上一個 <P> 的分斷。

假如你想用圖片連接一個檔案，也就是秀出圖片來，在滑鼠壓下圖片時，它會去連接另一個超文件，這時也許底下的例子會對你有所幫助：

```
<A HREF = "filename.htm"><IMG SRC="girls.gif"></A>
```

這個例子是說它會先秀出 girls.gif 的圖形，而圖形的外框有不一樣顏色的匡線圍繞，這表示它會連接另一個文件，此時滑鼠只要在圖形上按下，那超文件又會去連接另一個文件 (filename.htm) 了。

十四：外部圖形 External Image

也許你不想讓圖片在你的本文中出現，因為可能它不是必須的，或是檔案也許很大，或者是讀者可能沒有可以看圖片的工具，那在超文件中我們也可以讓讀

者以滑鼠按鍵的方式選擇看或不看圖片，只要他想看，就壓下滑鼠，然後超文件會開個 Session 來秀圖片，這種方式像你 load 另一篇超文件一般，它的方式如下：

```
<A HREF ="filename.gif"> 圖片 </A>
```

這時候圖片兩個字會變顏色，你只要壓下它，便可以看到圖片了。不過要注意的是他並不是秀在你的工具(如 Mosaic) 中。

十五： 其它要注意的地方

超文件的各種符號可以連續使用，當然它的語法必須要有層次性下面舉一個簡單的例子做說明：

```
<B><H1> This is Heading 1 </B></H1> ----- 錯
```

```
<B><H1> This is Heading 1 </H1></B> ----- 對
```

仔細瞧瞧上面這兩句的差別，你會發現超文件的語法是有層次的限制，假如每一種語法是一個回圈（譬如 和 之間。），也就是說必須由內圈到外圈一圈一圈的排列。不可以跳著排。

第六章

Moira 安裝說明

連仁豪

國立中央大學資訊管理學系三年級

im801012@im.mgt.ncu.edu.tw

6、1節 前言

在分散式網路環境中，對於網路資源的管理與維護是很重要的。尤其當環境中增加了許多的服務和周邊機器設備時，如何有效地管理與維護並且降低成本就顯得更為重要。此時，若能將各種資料集中處理而不是分別在各工作站上處理的話，就有可能減少管理與維護的費用。因此在分散式環境中，一個集中式的 **Service Management System** 是有其必要的。

6、2節 簡介

Moira 是 **MIT (Massachusetts Institute of Technology)**在發展 **Athena** 計劃時所設計而成的。由於在 **Athena** 的環境中，包括了數百部的各式各樣個人電腦與工作站，而且其使用者又為校園中的所有成員，再加上為數眾多的周邊設備，因此極需要一套方便又有效率的 **Service Management System**。所以 **MIT** 就開發了 **Moira** 這套系統，並在 1988 年正式應用在 **Athena** 的環境中。他包含了一個大的 **relational database**，一個前端的軟體去顯示資訊，和一個自動更新資料的更新系統。