

เดสก์ทอปลินุกซ์

เทพพิทักษ์ การุญบุญญานันท์



สารบัญ

1	ระบบ X Window	5
1.1	ระบบ X Client/Server	5
1.2	Window Manager	6
1.3	Desktop Environment	7
2	การปรับแต่ง GNOME	11
2.1	การติดตั้งฟอนต์	11
2.2	GConf	12
2.3	การแสดงผลตัวอักษร	13
2.4	พื้นหลัง	15
2.5	Theme	16
2.6	เมนู/ทูลบาร์	17
2.7	แป้นพิมพ์	18
2.8	เมาส์	20

ระบบ X Window

ระบบ GUI ที่อยู่คู่กับยูนิกซ์มานานคือระบบ X Window ซึ่งพัฒนาโดยโครงการ Athena ที่ MIT ร่วมกับบริษัท Digital Equipment Corporation และบริษัทเอกชนจำนวนหนึ่ง ปัจจุบัน X Window ดูแลโดย Open Group เป็นระบบที่เปิดทั้งในเรื่องโปรโตคอลและซอร์สโค้ด ขณะที่เขียนเอกสารฉบับนี้ เวอร์ชันล่าสุดของ X Window คือ เวอร์ชัน 11 รีลีส 6.6 (เรียกสั้นๆ ว่า X11R6.6)

สำหรับลินุกซ์และระบบปฏิบัติการในตระกูลยูนิกซ์ที่ทำงานบน PC ระบบ X Window ที่ใช้จะมาจากโครงการ XFree86 ซึ่งพัฒนาไดรเวอร์สำหรับอุปกรณ์กราฟิกต่างๆ ที่ใช้กับเครื่อง PC รุ่นล่าสุดขณะที่เขียนเอกสารนี้คือ 4.3.0

1.1 ระบบ X Client/Server

X Window เป็นระบบที่ทำงานผ่านระบบเครือข่าย โดยแยกเป็นส่วน X client และ X server สื่อสารกันผ่าน X protocol ดังนั้น โปรแกรมที่ทำงานบน X Window จะสามารถแสดงผลบนระบบปฏิบัติการที่ต่างชนิดกันก็ได้ ตราบใดที่ระบบนั้นสามารถให้บริการผ่าน X protocol ได้

X client ได้แก่โปรแกรมประยุกต์ต่างๆ ที่จะขอใช้บริการจาก X server ในการติดต่อกับฮาร์ดแวร์ เช่น จอภาพ แป้นพิมพ์ เมาส์ ฯลฯ ดังนั้น X server จึงทำงานอยู่บนเครื่องที่อยู่ใกล้ผู้ใช้เสมอ ในขณะที่ X client อาจอยู่ในเครื่องเดียวกันหรืออยู่ในเครื่องใดเครื่องหนึ่งในระบบเครือข่ายก็ได้

X client จะติดต่อกับ X server ด้วยการเรียก X library (เรียกสั้นๆ ว่า Xlib) API ต่างๆ ใน Xlib มีหน้าที่แปลงการเรียกฟังก์ชันแต่ละครั้งให้เป็น *request* ในรูปของ X protocol เพื่อส่งไปยัง X server ซึ่ง X server ก็จะได้รับ request ไปดำเนินการ และในทางกลับกัน เมื่อเกิดเหตุการณ์ต่างๆ กับฮาร์ดแวร์ เช่น ผู้ใช้กดปุ่มบนแป้นพิมพ์หรือลากเมาส์ X server ก็ส่ง *event* ผ่าน X protocol มายัง X client ซึ่ง event ก็จะถูกส่งเข้าไปต่อท้าย event queue ของ X client เพื่อรอ event loop ของ X client มารับไปดำเนินการต่อไป

ด้วยลักษณะเช่นนี้ ทำให้ X Window เป็นระบบที่ทำงานผ่านระบบเครือข่ายเสมอ โดยถ้า X client

และ X server ทำงานอยู่ในเครื่องเดียวกัน ก็จะสื่อสารกันผ่าน loopback interface (หมายเลข IP 127.0.0.1 หรือ localhost นั่นเอง)

1.2 Window Manager

หน้าที่ของ X server มีเพียงการให้บริการติดต่อฮาร์ดแวร์เท่านั้นจริงๆ จะไม่ทำหน้าที่ใดในระบบวินโดว์ต่างๆ ทั้งสิ้น หน้าที่ของการจัดสรรเนื้อที่บนหน้าจอให้กับ X client ต่างๆ รวมทั้งการวาดกรอบหน้าต่าง การ minimize และ maximize ฯลฯ จะเป็นหน้าที่ของโปรแกรมที่เรียกว่า *Window Manager*

Window manager ก็คือ X client หนึ่งที่ทำหน้าที่พิเศษในการเป็นตัวกลางจัดสรรพื้นที่หน้าจอ และทรัพยากรบางอย่างที่ใช้ร่วมกัน และมักจะมี interface สำหรับย้าย ย่อ ขยาย ปิดหน้าต่าง และสำหรับเรียกโปรแกรมต่างๆ ขึ้นมาทำงานด้วย โปรแกรมต่างๆ เมื่อต้องการเปิดหน้าต่าง จะร้องขอไปยัง window manager โดยบอกลักษณะหน้าต่างที่ต้องการไปด้วย แต่สุดท้ายจะได้หน้าต่างลักษณะไหนก็ขึ้นอยู่กับกติกาตัดสินใจของ window manager

ด้วยการทำงานที่เป็นคนละส่วนกับ X server เช่นนี้ ทำให้มีความเป็นไปได้ของรูปแบบของ window manager มากมาย และก็มีผู้สร้าง window manager ขึ้นมาหลากหลายมาก ในด้านหนึ่งก็ให้อิสระแก่ผู้ใช้ในการเลือกระบบ GUI ได้ตามรสนิยม ไม่จำเป็นต้องจำเจอยู่กับหน้าต่างแบบเดียว แต่ในอีกด้านหนึ่งก็ทำให้ผู้ใช้หน้าใหม่เกิดความละลานตาจนเลือกไม่ถูก ต่อไปนี้เป็นเพียงบางส่วนของ window manager ที่ถูกพัฒนาขึ้นในโลกของซอฟต์แวร์เสรี

twm หรือ Tab Window Manager เป็น window manager อย่างง่าย ไม่มีความหรูหรา ปรับแต่งไม่ได้ และไม่เปลืองทรัพยากร

fvwm95 เป็น window manager ที่มีหน้าต่างคล้าย Windows 95 เคยใช้เป็น window manager สำหรับลินุกซ์ยุคกลางๆ

olvwm หรือ OpenLook Virtual Window Manager พัฒนาเพิ่มเติมจาก olwm ที่ใช้เป็นมาตรฐานในระบบ XView บนระบบ Solaris จุดเด่นที่สังเกตง่ายคือไดอะล็อกที่มีเข็มหมุดปักติดกับเดสก์ทอปได้

AfterStep เป็น window manager ที่พัฒนามาจาก fvwm แต่ใช้หลักการ user interface ของ NeXTStep ซึ่งเป็นเครื่องเวิร์คสเตชันที่มีชื่อเสียงในยุคหนึ่ง จุดเด่นที่สังเกตง่ายคือ applet รูปสี่เหลี่ยมจัตุรัสที่วางเรียงอยู่ตามขอบของเดสก์ทอป ซึ่งใช้แทนทั้ง shortcut สำหรับเรียกโปรแกรม และแทนโปรแกรมที่ minimize รวมทั้งบานหน้าต่างที่มีเอกลักษณ์เฉพาะตัวของ NeXTStep และภาพกราฟิกคุณภาพสูง

WindowMaker เป็น window manager อีกตัวหนึ่งที่ใช้ user interface แบบ NeXTStep แนวคิดเรื่อง interface จึงคล้ายกับ AfterStep เป็น window manager ที่เขียนขึ้นใหม่เกือบทั้งหมด มีจุดมุ่งหมายให้เป็น window manager ที่ขนาดเล็ก ไม่เปลืองทรัพยากร ปรับแต่งง่ายโดยไม่ต้องแก้ไขไฟล์ configuration เป็นอีกโครงการหนึ่งที่มีภาพกราฟิกคุณภาพสูงตั้งแต่รุ่นมาตรฐาน

ICEWM สามารถปรับแต่งให้คล้าย Windows 95 หรือ OS/2 หรือ Motif ได้ เน้น interface ที่สวยงามแบบเรียบง่าย ขนาดเล็ก

Enlightenment เป็น window manager ที่มีจุดเด่นเฉพาะตัวในเรื่องลูกเล่นกราฟิก เสียง และการปรับแต่งที่ให้อิสระแก่ศิลปินอย่างเต็มที่ มี applet (เรียกว่า epplet) เหมาะสำหรับผู้นิยมปรับแต่งเดสก์ทอปและต้องการดึงความสามารถต่างๆ ของระบบขึ้นมาใช้อย่างเต็มที่ เคยใช้เป็น window manager มาตรฐานของ GNOME มาก่อน ก่อนที่ GNOME จะเริ่มย้ายแนวคิดเรื่องระบบปรับแต่งเดสก์ทอปไปที่ desktop environment แทนที่จะหนักที่ window manager โดยเริ่มใช้ Sawfish ที่มี feature น้อยกว่าแทน

Sawfish ออกแบบให้สามารถปรับแต่งได้โดยใช้ภาษา LISP บรรยายธัม ทำให้สามารถปรับแต่งหน้าตาของหน้าต่างได้ทุกอย่าง เคยใช้เป็น window manager มาตรฐานของ GNOME 1.4 มาก่อน และปัจจุบันกำลังย้ายมา GNOME 2

Metacity ใช้ GTK+ 2 เป็นหลัก ออกแบบมาเพื่อการทำงานร่วมกับ GNOME 2 อย่างเต็มที่ ไม่ว่าจะเป็นการปรับแต่งหรือระบบธัม จึงเป็น window manager มาตรฐานสำหรับ GNOME 2

KWin เป็น window manager หลักสำหรับ KDE

BlackBox เป็น window manager ขนาดเล็ก เบา ใช้ทรัพยากรน้อยกว่าระดับที่ Window Maker ใช้ แต่ยังสามารถปรับแต่งหน้าต่างให้สวยงามได้ พัฒนาด้วย C++ แต่ค่อนข้างจะไม่เหมาะสำหรับผู้ที่ต้องการความคล้ายกับ Windows หรือ OSX

OpenBox พัฒนาต่อจาก BlackBox เพิ่มความสามารถเรื่องฟอนต์ลรอยหยัก แต่ยังคงความเบาไว้เหมือนเดิม

FluxBox พัฒนาต่อจาก BlackBox ความสามารถต่างๆ ที่เพิ่มขึ้นมาคือการจัดกลุ่มหน้าต่างเป็นแท็บซ้อนกันเหมือนที่คั่นหนังสือได้

1.3 Desktop Environment

แม้ยูนิกซ์จะมีกลไก inter-process communication (IPC) ต่างๆ ทำให้โปรแกรมต่างๆ สามารถทำงานร่วมกันได้ ไม่ว่าจะเป็ socket, pipe, semaphore, shared memory, X protocol, Remote Procedure Call (RPC) ฯลฯ แต่บนระบบเดสก์ทอปในยุคหนึ่งกลับขาดกลไกการเชื่อมโยงที่ออกแบบเพื่องานเดสก์ทอปโดยเฉพาะ โปรแกรมบน X Window ทั้งหลายมีอิสระเต็มที่ที่จะใช้กลไกใดก็ได้ ทำให้ความหลากหลายกลายเป็นความไม่มีมาตรฐานร่วมกัน ในขณะที่วินโดวส์มีการพัฒนาระบบ DCOM ขึ้นมานั้น ทางมาตรฐานของ OMG (Object Management Group) ก็มี CORBA (Common Object Request Broker Architecture) ที่ทำงานคล้ายกัน คือเป็นคล้าย RPC ในภาค Object-Oriented แต่ในขณะเดียวกัน วินโดวส์ได้ก้าวล้ำไปหนึ่งขั้น คือการสร้างกลไก OLE (Object Linking and Embedding) เพื่อสนับสนุนเอกสารที่สามารถฝัง object ของโปรแกรมชนิดอื่นได้ (ความจริงแล้วตัวกลไกของยูนิกซ์และ X Window ก็สามารถทำเช่นนั้นได้ เพียงแต่ยังไม่มีการพัฒนา framework สำเร็จรูปโดยเฉพาะเท่านั้น)

เมื่อเป็นเช่นนี้ สภาพแวดล้อมแบบเดิมของ X Window ที่ปล่อยให้ทุกอย่างเป็อิสระจึงไม่เอื้ออำนวยต่อการทำงานร่วมกันบนเดสก์ทอป จึงได้เกิดความพยายามที่จะสร้าง *Desktop Environment* เพื่อกำหนดข้อตกลงร่วมกันระหว่างโปรแกรมต่างๆ ตั้งแต่การเลือก toolkit (ไลบรารีสำหรับสร้างปุ่ม เมนู และองค์ประกอบ GUI ต่างๆ) เพื่อให้รูปลักษณ์ของทุกโปรแกรมปรากฏเป็นชุดเดียวกัน ไปจนถึงการกำหนดโครงสร้างของการทำงานร่วมกันบนเดสก์ทอป

ความพยายามแรกเกิดขึ้นที่ Open Group เอง โดยร่วมกับ Sun สร้าง CDE (Common Desktop Environment) โดยอาศัย Motif เป็น toolkit แต่ CDE ก็ยังไม่เป็นที่นิยมเท่าที่ควร เพราะรูปแบบการพัฒนาที่ค่อนข้างปิด (ไม่ใช่ซอฟต์แวร์เสรี) และมีใช้เฉพาะบางแพลตฟอร์มเท่านั้น จึงเกิดกลุ่มใหม่ในฝ่ายซอฟต์แวร์เสรีขึ้น คือ KDE (K Desktop Environment) โดยใช้ Qt เป็น toolkit และพัฒนาระบบ KPart สำหรับ embed object ผ่าน CORBA

อย่างไรก็ดี ปัญหาของ KDE ในขณะนั้นคือ Qt ที่มี license แปลกประหลาดไม่เป็นที่ยอมรับในวงการซอฟต์แวร์เสรีบางส่วน ทั้งนี้เพราะบริษัท TrollTech ผู้ผลิตยังต้องการทำธุรกิจกับ Qt อยู่นั่นเอง กลุ่มซอฟต์แวร์เสรีส่วนที่นิยมอดมคติจึงได้ตั้งโครงการใหม่ที่ใช้ license แบบ GNU อย่างแท้จริง คือ GNOME (GNU Network Object Management Environment) โดยเลือก GTK+ (GIMP Tool Kit) เป็น toolkit หลัก

ปัจจุบัน KDE และ GNOME จึงกลายเป็นระบบเดสก์ทอปหลักในโลกซอฟต์แวร์เสรี ทั้งสองกลุ่มมีการพัฒนาเป็นอิสระแยกจากกัน แม้จะมีการใช้ทรัพยากรบางอย่างร่วมกันบ้าง และมีการติดต่อสื่อสารระหว่างนักพัฒนาทั้งสองค่ายบ้าง แต่ตัวซอฟต์แวร์ต่างๆ เป็นคนละชุดกัน

KDE นั้น พัฒนาด้วยภาษา C++ เป็นหลักตั้งแต่ Qt ขึ้นมา และภายหลังได้แก้ปัญหาเรื่อง license ของ Qt แล้ว แต่ GNOME ก็ยังพัฒนาต่อไป เนื่องจากได้พัฒนาไปมากแล้ว มีฐานผู้ใช้มากแล้วในขณะนั้น ภาษาที่ใช้พัฒนาโครงสร้างพื้นฐานของ GNOME คือภาษา C แต่การออกแบบก็เป็นแบบ object-oriented และมีการ bind กับภาษาต่างๆ มากมาย เช่น C++, Objective-C, Ada, Pascal, Python, Ruby ฯลฯ

นอกจากโครงสร้างพื้นฐานแล้ว เดสก์ทอปทั้งสองยังได้พัฒนาโปรแกรมประยุกต์ต่างๆ บนโครงสร้างดังกล่าวด้วย ดังตารางที่ 1.1 แสดงการเปรียบเทียบองค์ประกอบต่างๆ ของเดสก์ทอปทั้งสอง

ตารางที่ 1.1: เปรียบเทียบองค์ประกอบของ KDE และ GNOME

องค์ประกอบ	KDE	GNOME
Toolkit	Qt	GTK+
Component	KPart	bonobo
Display Manager	KDM	GDM
Text Editor	kedit, kwrite, kate	gedit
Word Processor	KWord	AbiWord
Spreadsheet	KSpread	gnumeric
Paint	KPaint	The GIMP
Draw	Kontour	Sodipodi
Diagram	Kivio	Dia
Presentation	KPresent	Agnubis (หยุดการพัฒนา)
Developer's IDE	KDevelop	Anjuta
Project Management	?	MrProject
Financial	KMyMoney2, KBudget	GnuCash
Web Browser	Konqueror	Mozilla, Galeon, Epiphany
Web Editor	Quanta+	Bluefish
Mail	KMail	Balsa, Evolution
News	KNode	Pan
IRC	kIRC	xchat

ICQ	licq	GnomeICU
AIM	Kinkatta	gaim
Yahoo! Messenger	KYIM	?
MSN Messenger	kmerlin	?
Image Viewer	Pixie	GQView, gThumb, eog
Media Player	noatun	xmms
Movie Player	Kaffeine	Totem
Sound Editor	KWave	Beast
Digital Camera	Digikam, Kamera	gPhoto

การปรับแต่ง GNOME

2.1 การติดตั้งฟอนต์

ระบบฟอนต์บน GNOME 2 และ KDE 3 จะใช้ fontconfig เป็นหลัก ซึ่งไฟล์ที่เก็บค่าต่างๆ ของ fontconfig ได้แก่ `/etc/fonts/fonts.conf` ซึ่งนอกจากจะเก็บที่อยู่ของฟอนต์แล้ว ยังเก็บข้อกำหนดการ match ชื่อฟอนต์ต่างๆ อีกด้วย

อย่างไรก็ตาม กลไกของ fontconfig ได้ออกแบบให้เอื้อต่อการติดตั้งฟอนต์มาก คุณแทบไม่ต้องแตะต้อง config file ดังกล่าวเลย ถ้าติดตั้งฟอนต์ไว้ที่ใดเรกทอรีใดใดเรกทอรีหนึ่งได้ราก็กำหนดคือ `/usr/share/fonts` หรือ `/usr/X11R6/lib/X11/fonts` หรือ `~/.fonts` นอกเสียจากคุณจะมีการกำหนดการ match ชื่อฟอนต์ด้วย เช่น เมื่อต้องการให้ชื่อ “sans-serif” ใช้ฟอนต์ “Garuda” (เมื่อแสดงภาษาไทย) คุณก็เพิ่มรายการใน `<alias>` ดังตัวอย่าง

```
<alias>
  <family>sans-serif</family>
  <prefer>
    <family>Verdana</family>
    <family>Nimbus Sans L</family>
    <family>Luxi Sans</family>
    <family>Garuda</family>
    <family>Arial</family>
    <family>Helvetica</family>
    <family>Kochi Gothic</family>
    <family>AR PL KaitiM GB</family>
    <family>AR PL KaitiM Big5</family>
    <family>Baekmuk Dotum</family>
```

```
<family>SimSun</family>
</prefer>
</alias>
```

fontconfig จะมีระบบ cache สำหรับรายชื่อฟอนต์ในไดเรกทอรี โดยอาศัยไฟล์ `fonts.cache-1` ที่แต่ละไดเรกทอรีของฟอนต์ ซึ่งโดยปกติ ไลบรารี fontconfig จะมีการปรับปรุง cache อยู่เป็นระยะๆ เมื่อมีการเพิ่มฟอนต์ใหม่ ทำให้โปรแกรมต่างๆ ทราบได้ แต่เพื่อความมั่นใจ คุณจึงควรสั่งสร้าง cache ใหม่ทันทีที่ติดตั้งฟอนต์ โดยใช้คำสั่ง `fc-cache` ซึ่งมีรูปแบบ

```
fc-cache [-f] [-v] [directory]
```

ตัวเลือก `-f` (force) เป็นการบังคับให้สร้าง cache เสมอโดยไม่ต้องตรวจสอบ ส่วนตัวเลือก `-v` (verbose) จะทำให้ `fc-cache` แสดงชื่อไดเรกทอรีที่กำลังสร้าง cache ด้วย คุณสามารถระบุให้สร้าง cache เฉพาะไดเรกทอรีที่ต้องการได้โดยกำหนดชื่อ *directory* ในบรรทัดคำสั่ง ซึ่งหากไม่กำหนด จะหมายความว่าสร้าง cache ใหม่ทั้งหมด

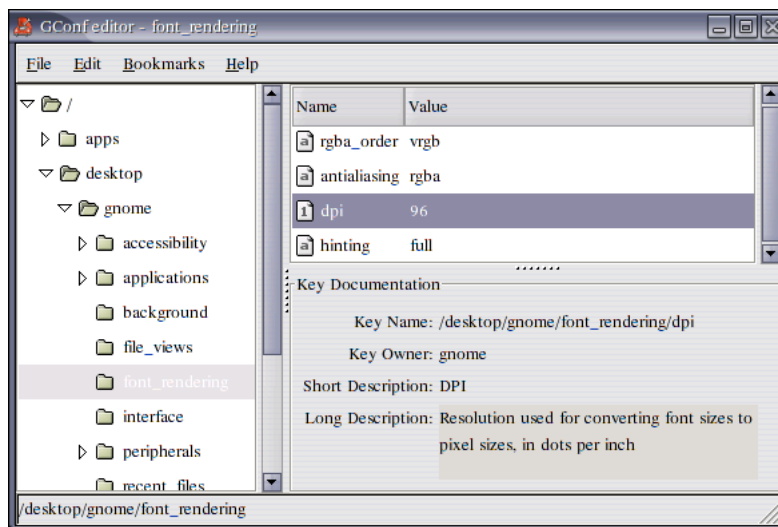
สมมติว่าคุณเพิ่งติดตั้งฟอนต์ใหม่ไว้ที่ `/usr/share/fonts/th/TTF` คุณอาจสั่ง

```
# fc-cache -fv /usr/share/fonts/th/TTF
fc-cache: "/usr/share/fonts/th/TTF/": caching, 12 fonts, 0 dirs
fc-cache: succeeded
```

คุณสามารถเรียกดูรายการฟอนต์ทั้งหมดใน cache ของ fontconfig ได้ด้วยคำสั่ง `fc-list`

2.2 GConf

GNOME 2 มีการเปลี่ยนแปลงระบบ configuration จาก GNOME 1.4 เดิมที่โปรแกรมต่างๆ เคยกำหนด configuration file ของใครของมัน ก็เปลี่ยนมาใช้ระบบ configuration ร่วม ซึ่งมีแนวคิดคล้ายกับ registry ของวินโดวส์ แต่รูปแบบการเก็บข้อมูลจะเป็น XML และมีการแยกข้อมูลเป็นสัปดาห์เป็นส่วน ผู้ใช้แต่ละคนที่ login เข้าใช้ GNOME จะมี configuration directory ของตัวเองอยู่ที่ `~/.gconf` แต่ไฟล์ต่างๆ ในไดเรกทอรีนี้ไม่ได้ออกแบบให้ผู้ใช้แก้ไขโดยตรงด้วย editor แต่จะเขียนผ่านเครื่องมือปรับแต่งต่างๆ หรือในระดับต่ำที่สุดของการแก้ไข config ก็คือการใช้โปรแกรม `gconf-editor` ซึ่งคล้ายกับการใช้ `regedit` ในวินโดวส์นั่นเอง



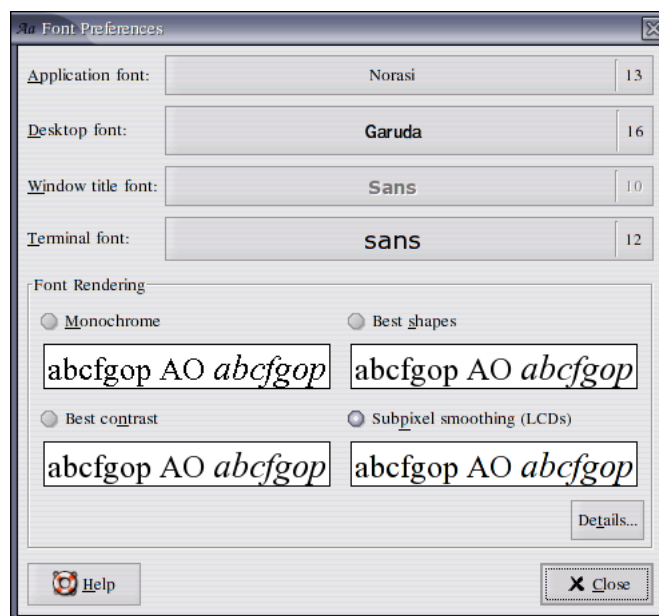
กลไกของระบบ config ของ GNOME นี้เรียกว่า GConf ซึ่งเป็นการปรับแต่งแบบทันที คือจะมีผลทันทีที่เปลี่ยนค่า ทั้งนี้เพราะจะมีการแจ้งไปยังโปรแกรมต่างๆ ให้ update ตัวเองนั่นเอง

หัวข้อต่างๆ ที่จะกล่าวถึงต่อไปนี้เป็น การปรับแต่ง GNOME ผ่านระบบ GConf ด้วยโปรแกรมที่อยู่ใต้เมนู “Desktop Preferences” หรือ “ปรับแต่งพื้นโต๊ะ”

2.3 การแสดงตัวอักษร

คุณสามารถเลือกฟอนต์สำหรับใช้ในส่วนต่างๆ ของเดสก์ทอป และปรับคุณภาพการแสดงผลตัวอักษร ด้วยการลรอยหยักได้ โดยเลือกเมนู

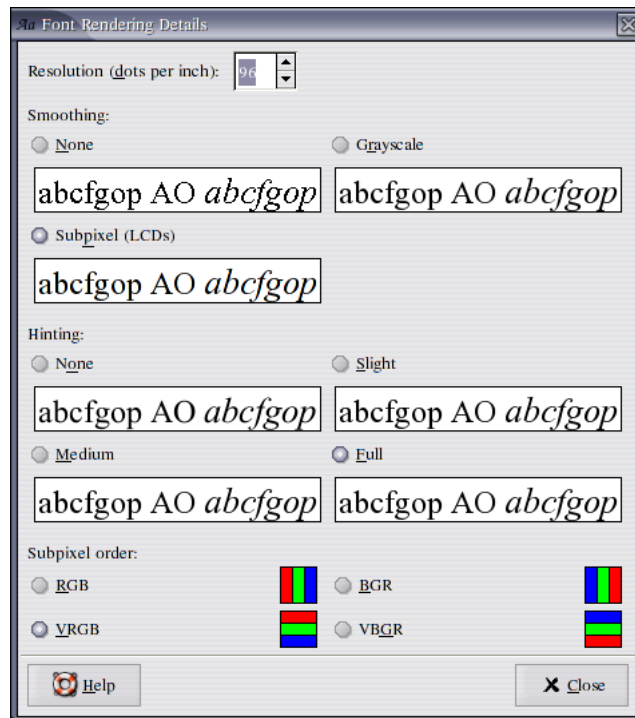
Desktop Preferences -> Fonts



ในส่วนของการปรับคุณภาพการแสดงผลนั้น คุณสามารถเลือกวิธีแสดงผลได้ 4 แบบ

- “**M**onochrome” ไม่ลบรอยหยัก สีที่ใช้แสดงจะมีเพียงขาวกับดำเท่านั้น
- “**B**est **c**ontrast” ลบรอยหยักด้วย grayscale โดยเกลี่ยสีแคบๆ เพื่อให้ได้ขอบคมที่สุด
- “**B**est **s**hapes” ลบรอยหยักด้วย grayscale โดยเกลี่ยสีกว้างเล็กน้อย เพื่อให้ได้รูปร่างตัวอักษรที่ใกล้เคียงที่สุด

- “Subpixel smoothing (LCDs)” ลบรอยหยักด้วยสีแดง เขียว น้ำเงิน ซึ่งจะให้คุณภาพดีสำหรับจอ LCD ในการลบรอยหยักด้วยสีนี้ ยังสามารถเลือกการใช้สีได้อีก 4 แบบ คือไล่สี RGB ซ้ายไปขวา, ขวาไปซ้าย, บนลงล่าง หรือล่างขึ้นบน คุณเลือกได้ (พร้อมการกำหนดรายละเอียดทางเทคนิคอื่นๆ เช่น การใช้ hint) โดยกดปุ่ม “Details. . .”



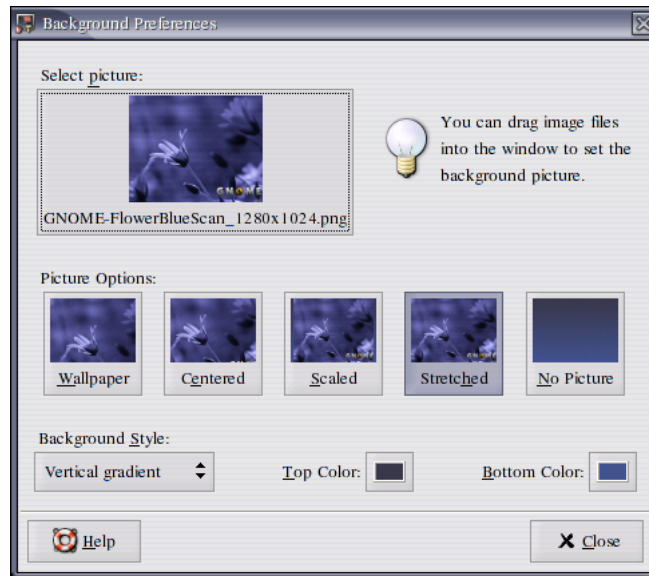
2.4 พื้นหลัง

เมนูที่ใช้คือ

Desktop Preferences -> Background

หรือ context menu (ด้วยเมาส์ปุ่มขวา) ที่พื้นเดสก์ทอป

คุณสามารถเลือกใช้สีพื้นหรือใช้รูปภาพเป็นพื้นหลังได้ โดยสามารถกำหนดวิธีวางรูปภาพได้ด้วย

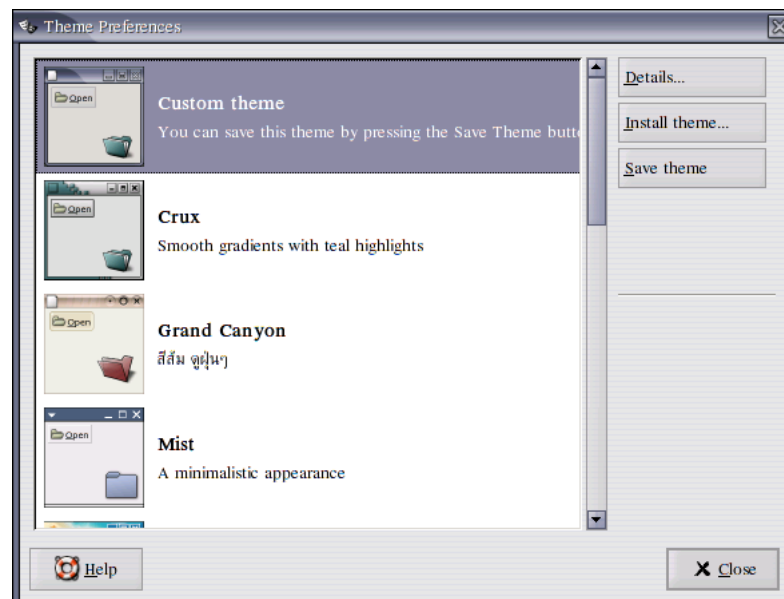


2.5 Theme

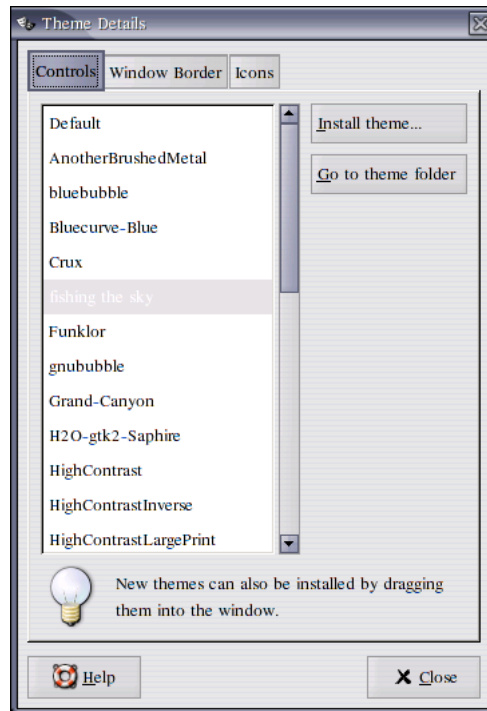
ระบบ theme ของ GNOME จะแบ่งเป็นสามส่วน คือ

1. GTK+ theme ซึ่งจะมีผลต่อ widget ต่างๆ
 2. Window Manager theme ซึ่งจะมีผลกับกรอบหน้าต่าง
 3. Icon theme ซึ่งจะมีผลต่อภาพ icon ในเมนูและใน shortcut ต่างๆ
- เมนูสำหรับปรับเปลี่ยน theme คือ

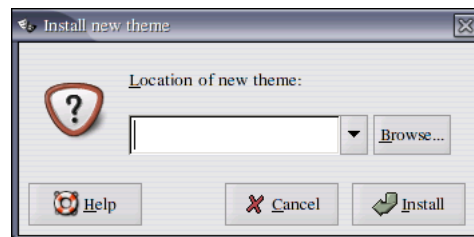
Desktop Preferences -> Theme



จะแสดง theme ในลักษณะของ metatheme คือรวม theme ทุกส่วนเข้าเป็นชุด คุณสามารถขอดูรายละเอียดแต่ละส่วนของ theme ได้โดยกดปุ่ม “Details. . .”



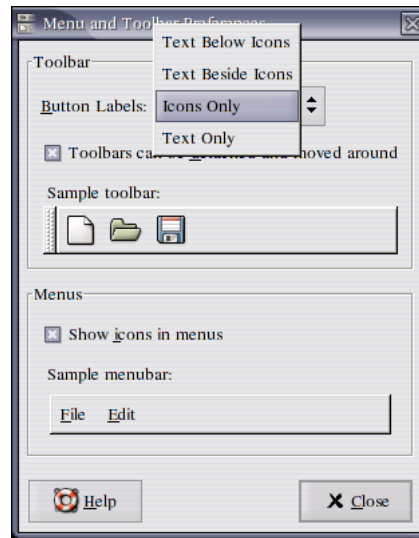
หรือกดปุ่ม “Install theme...” เพื่อติดตั้ง theme (ดาวน์โหลดได้จาก <http://arts.gnome.org>)



2.6 เมนู/ทูลบาร์

คุณสามารถปรับแต่งลักษณะปรากฏของเมนูและทูลบาร์สำหรับโปรแกรม GNOME ได้ด้วยเมนู

Desktop Preferences -> Menus & Toolbars

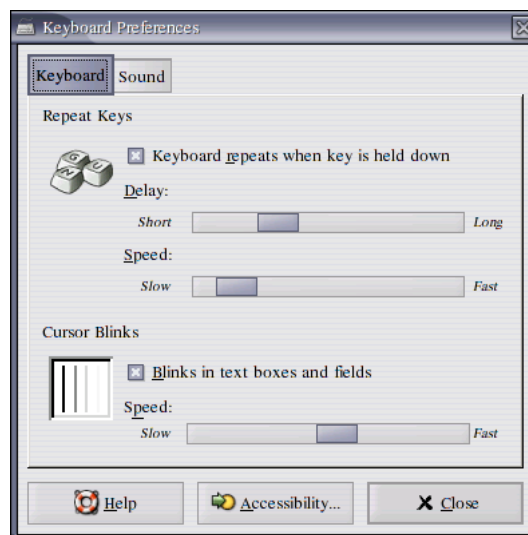


คุณสามารถเลือกให้ฉีกเมนูและทูลบาร์ออกมาวางข้างนอกได้หรือไม่ได้ รวมทั้งรูปแบบของรายการเมนูและทูลบาร์ว่าจะแสดง icon และข้อความกำกับอย่างไร

2.7 แป้นพิมพ์

คุณสามารถตั้งคุณสมบัติของแป้นพิมพ์ได้ในเมนู

Desktop Preferences -> Keyboard

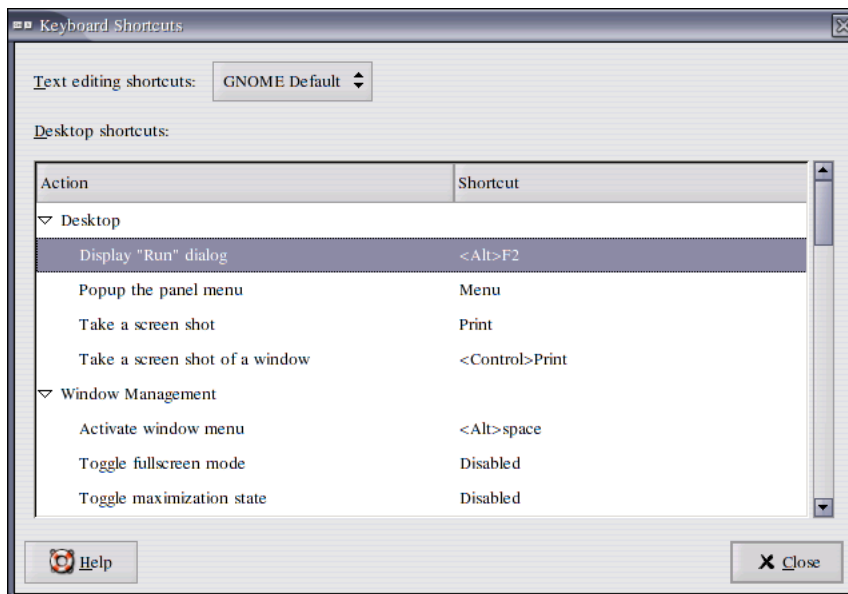


ซึ่งจะมีสองส่วน ส่วนแรกเป็นอัตราการซ้ำปุ่มเดิมเมื่อกดค้างและอัตราการกระพริบของเคอร์เซอร์ ส่วนที่สอง (Sound) เป็นการเปิด/ปิดเสียงบี๊ป



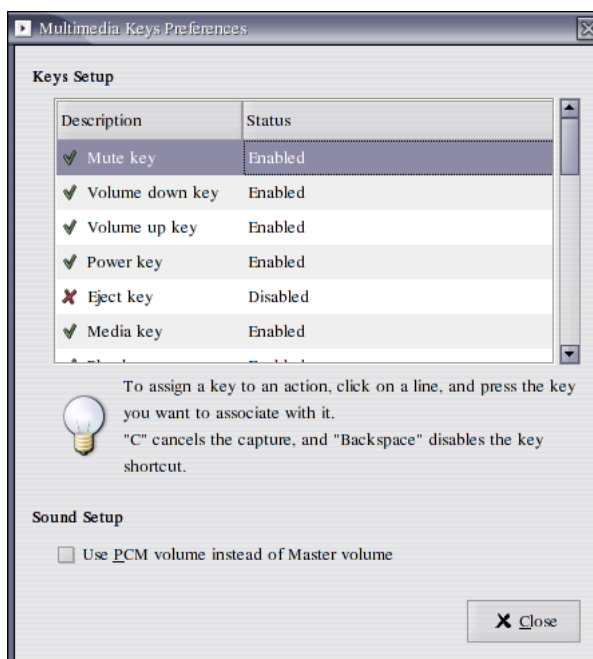
เมนูถัดมาที่เกี่ยวข้องกับแป้นพิมพ์เป็นการตั้งปุ่มลัด โดยเลือกเมนู

Desktop Preferences -> Keyboard Shortcuts



คุณสามารถเลือกรายการที่ต้องการแล้วกดปุ่มลัดเพื่อกำหนดค่าได้ตามต้องการ
 และถ้าคุณใช้แป้นพิมพ์ที่มีปุ่มอินเทอร์เน็ทและมัลติมีเดีย คุณสามารถกำหนดหน้าที่ของปุ่มแต่ละปุ่ม
 ได้ผ่านเมนู

Desktop Preferences -> Multimedia Keys



ลักษณะการตั้งปุ่มลัดก็จะคล้ายกัน คือเลือกรายการที่ต้องการแล้วกดปุ่มลัดเพื่อกำหนดค่า แต่โปรแกรมจะไม่แสดงค่าของปุ่มปัจจุบัน เพราะยังไม่มีมาตรฐานสำหรับชื่อปุ่มเหล่านี้

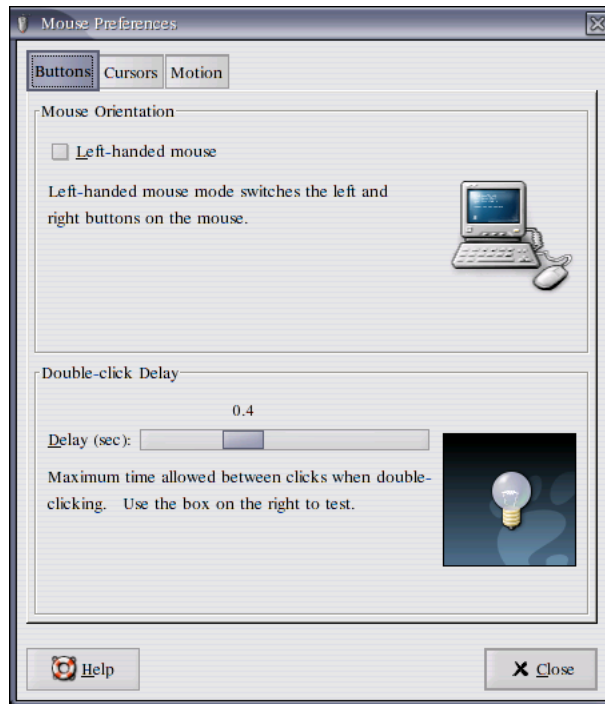
แต่ทั้งนี้ คุณจะต้องเลือกแบบแป้นพิมพ์ที่เหมาะสมใน XF86Config-4 ด้วย จึงจะใช้ปุ่มพิเศษต่างๆ นี้ได้

2.8 เม้าส์

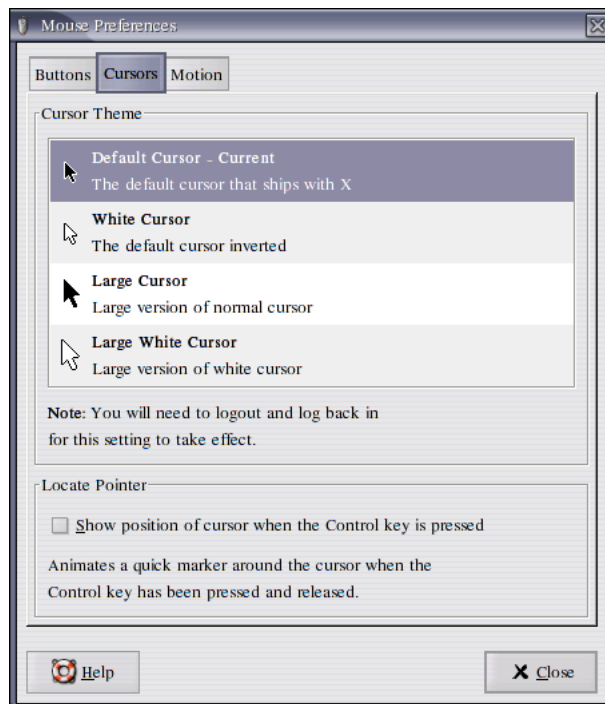
คุณสามารถปรับแต่งพฤติกรรมของเม้าส์และเคอร์เซอร์ได้ผ่านเมนู

Desktop Preferences -> Mouse

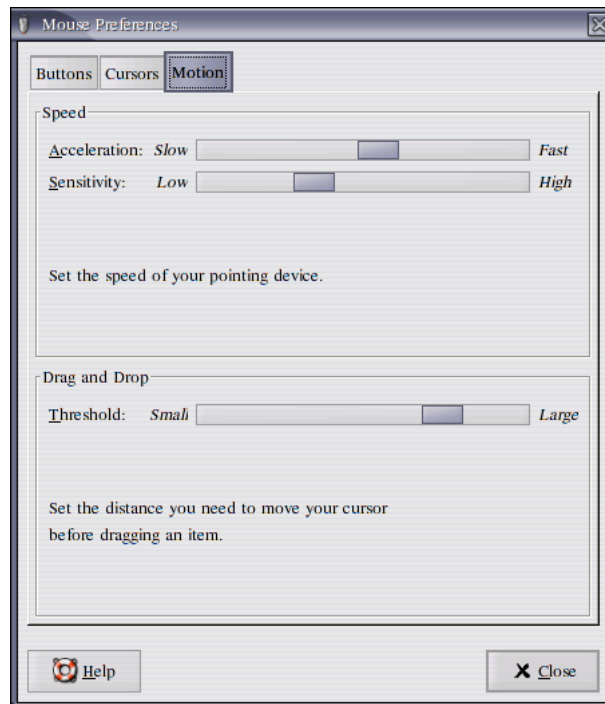
ซึ่งจะมีส่วนกำหนดค่าสามส่วน คือปุ่มเม้าส์ เคอร์เซอร์ และความไวของการเลื่อนเม้าส์



คุณสามารถสลับปุ่มเม้าส์ซ้าย-ขวาได้ ถ้าคุณถนัดซ้าย และคุณสามารถกำหนดช่วงห่างของ double-click ได้ โดยสามารถทดสอบได้ที่ภาพหลอดไฟ



คุณสามารถเลือกเคอร์เซอร์เมาส์ได้หลายแบบ และสามารถเลือกให้ใช้ปุ่ม **Ctrl** เพื่อแสดงตำแหน่งเมาส์ (ใช้ในกรณีที่มีเมาส์หลายตัว) ได้ด้วย



ในส่วนของความไวของการเลื่อนเมาส์ คุณสามารถตั้งความละเอียดของการเลื่อนเมาส์ได้ และสามารถตั้งขอบเขตของการเคลื่อนไหวก่อนที่จะเริ่มถือว่าการ drag ได้ด้วย