



# **Preparación para el examen LPI 101**

## **Tema 110**

### **El sistema X-Window**

## **Créditos y licencia de uso**

### **Coordinación:**

Manuel Guillán (xLekOx) [lpi@xleko.org](mailto:lpi@xleko.org)

### **Traducción:**

Juan Maria Gil (Smooth) [yo@juanmaria.com](mailto:yo@juanmaria.com)

Dani Donisa (kasei) [kasei@flashmail.com](mailto:kasei@flashmail.com)

### **Maquetación:**

Manuel Guillán (xLekOx) [lpi@xleko.org](mailto:lpi@xleko.org)

Versión 1.0 (07-03-2005 14:00)

Distribuido por FreeUOC ([www.freeuoc.org](http://www.freeuoc.org)) bajo licencia: Attribution-NonCommercial-ShareAlike2.0 de commons creative



<http://creativecommons.org/licenses/by-nc-sa/2.0/>

## ÍNDICE

### Índice de contenido

Tema 110	
El sistema X-Window.....	1
Créditos y licencia de uso.....	2
ÍNDICE.....	3
Tema 110.1	
Instalando y configurando el entorno gráfico.....	4
Introducción.....	5
El entorno X.....	6
Requerimientos de Hardware.....	6
Instalando Xfree86.....	7
Configurando el sistema X Window.....	8
El fichero XF86Config.....	11
Controlando X.....	15
Iniciando X.....	16
Tema 110.2	
Configurando el inicio de sesiones gráficas.....	17
Introducción.....	19
Configurando el proceso de inicio.....	20
Los arranques gráficos: xdm.....	22
Activando y desactivando XDM.....	22
Configurando XDM.....	22
KDM y GDM.....	23
Tema 110.4	
Configurando el administrador gráfico.....	24
Introducción.....	25
Arrancando X.....	26
Arranque manual de X.....	26
Los ficheros xinitrc y .xinitrc.....	27
Los ficheros Xclients y .Xclients.....	27
Utilizando X.....	27
Utilizando un emulador de terminal.....	29
Personalizando las aplicaciones X.....	30
Utilizando las teclas especiales.....	31
Gestionando aplicaciones con mal comportamiento.....	31
Ejecutando Remotamente X y Clientes.....	32
Configurando la seguridad de X.....	32
Configurando Clientes Remotos.....	33
Bibliografía y enlaces recomendados.....	35

# **Tema 110.1**

## **Instalando y configurando el entorno gráfico**

## **Introducción**

En este capítulo se verá como instalar y configurar el entorno gráfico de los sistemas GNU/Linux, los proyecto Xfree86 y x.org (más actualmente) son los principales proveedores. Se verá como se estructura el fichero de configuración, monitor, tarjeta gráfica, fuentes...

Nota: Este tema puede tener contenidos que a fecha de hoy estén desfasados o cambiados.

Este tema tiene un peso (importancia) de 5 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

## El entorno X

El sistema X Window fue desarrollado como el componente de visualización del Proyecto Athena en el MIT. El sistema X Window es un entorno gráfico amplio y potente para los sistemas UNIX. Desde que aparecieron las primeras versiones de X Window muchos distribuidores de UNIX la incorporaron a sus plataformas. La amplia disponibilidad de X Window en Unix ha hecho que se convierta en la interfaz gráfica estándar para los sistemas Unix. Prácticamente todos los sistemas Unix del mundo ruedan alguna variedad del sistema X Window.

La implementación del sistema X Window en Linux se basa en la adaptación distribuíble de X Window versión 11 release 6 (habitualmente conocida como X11R6). Esta adaptación de libre distribución se conoce como *Xfree86* para las familias de procesadores 08386/80486 y Pentium. Desde su adaptación inicial, Xfree86 ha estado disponible para otras plataformas incluyendo System V/386, 386BSD.

La implementación Xfree86 incluye todos los fuentes, binarios, librerías y herramientas.

En primer lugar es necesario aclarar alguna terminología. El servidor X es el controlador de visualización y se encarga de controlar la pantalla en la estación de trabajo local. Esto contrasta con la definición típica de *servidor*, que es un recurso que suele encontrarse en un sistema remoto.

Los clientes X son los programas aplicativos como el administrador de ficheros o el manejador de ventanas (por ejemplo KDE), con los que el usuario interactúa. Al contrario que el servidor que se encuentra en el equipo local, un cliente puede rodar tanto en el sistema local como en uno remoto.

## Requerimientos de Hardware

El sistema X Window está muy ligado al hardware de vídeo del sistema y una configuración incorrecta del servidor X podría dañar el monitor o la tarjeta de vídeo. No obstante, la mayoría de los monitores multisync actuales son bastante resistentes a este tipo de averías ya que son capaces de ajustarse automáticamente al sincronismo y frecuencia de refresco del adaptador. De todas formas es conveniente tener cuidado al modificar este tipo de ajustes.

Durante la instalación de Linux, el programa de instalación de X Window detecta el hardware de vídeo disponible y solicita verificación de dicha selección al usuario. Especialmente importante es el chipset de la tarjeta de vídeo que determinará que servidor X se utilizará.

**Nota:** Existe una lista de hardware soportado con los correspondientes chipsets en [www.xfree86.org/4.0/Status.html](http://www.xfree86.org/4.0/Status.html).

Cada fabricante de adaptadores de vídeo utiliza un determinado chipset responsable de las funcionalidades gráficas del producto. Este chipset está indicado normalmente en la documentación del adaptador pero, en ocasiones, será necesario contactar con el fabricante para poder averiguarlo.

Para rodar X Window en un sistema Linux es necesario, **como mínimo**, un sistema 80486 con un adaptador de vídeo compatible con un mínimo de 8MB de RAM. Para unas prestaciones óptimas se recomienda más RAM y una tarjeta con acelerador de vídeo.

En todo caso, antes de comprar o intentar configurar un adaptador de vídeo es conveniente verificar su compatibilidad consultando la información de [www.xfree86.org/4.0/Status.html](http://www.xfree86.org/4.0/Status.html).

También se recomienda añadir RAM al PC, ya que un sistema con 4MB de RAM puede ser hasta diez veces más lento que uno con 8MB (debido al swapping). El mínimo ideal para un sistema X Window system sería 16MB de RAM.

## **Instalando Xfree86**

La mayoría de las distribuciones de Linux incluyen el sistema X Window con pequeñas diferencias en la instalación dependiendo del proveedor. RedHat por ejemplo utiliza un programa de instalación para su sistema RedHat 7 totalmente basado en X y, por tanto, el usuario ha de escoger y chequear la configuración de X Window antes de comenzar la instalación del sistema. Este método apenas requiere de configuración adicional a no ser que haya que configurar alguna función o prestación específica.

Generalmente no habrá que descargar los fuentes de Xfree86 por separado porque normalmente se instalan durante la instalación de Linux.

Se pueden encontrar distribuciones binarias de Xfree86 para Linux en varios servidores FTP. (P.Ej. en el sitio FTP XFree86: <ftp://ftp.xfree86.org/pub/XFree86/4.0.2/binaries/>).

También serán necesarios los subdirectorios Linux-axp-glibc21, Linux-ix86-glibc20, o Linux-ix86-glibc21, dependiendo del tipo de procesador (axp=alpha, ix86=Intel80x86) y de la versión de la librería Gnu C.

Antes de hacer nada más, se debe descargar y ejecutar el script Xinstall.sh, porque de esta forma conoceremos cualquier requisito necesario que haya que cumplir antes de continuar con la instalación.

Las instrucciones completas para la instalación de los binarios de Xfree86 se encuentran en [www.xfree86.org/4.0/Install2.html](http://www.xfree86.org/4.0/Install2.html).

Para descargar los ficheros es mejor utilizar FTP ya que hay que descargar bastantes ficheros.

Por ejemplo, al ejecutar Xinstall.sh podríamos obtener los siguientes resultados:

```
[root@poweredge chare]# sh Xinstall.sh
Welcome to the XFree86 4.0 installer
You are strongly advised to backup your existing XFree86
installation before proceeding. This includes the /usr/X11R6
and /etc/X11 directories. The installation process will overwrite
existing files in those directories, and this may include some
configuration files that may have been customised.

Do you wish to continue? (y/n) [n] y

Checking which OS you're running...
uname reports 'Linux' version '2.2.16-22smp', architecture 'i586'.
Object format is 'ELF'. libc version is '6.1'.
```

Checking for required files ...

You need to download the 'extract' (or 'extract.exe') utility and put it in this directory.

When you have corrected the problem, please re-run 'sh Xinstall.sh' to proceed with the installation.

En este ejemplo, el script Xinstall.sh script identifica algunos componentes que faltan, en particular, el programa de extracción. Lo único que hace falta para instalar XFree86 es conseguir la distribución binaria, crear el directorio /usr/X11R6 (como root), y desempaquetar los ficheros desde /usr/X11R6.

Una vez que los ficheros estén desempaquetados en /usr/X11R6, hay que añadir /usr/X11R6/bin al path de búsqueda editando el .profile o añadiéndolo al entorno actual, pero de esta última forma se perderán los cambios al desconectarse del sistema.

Aunque se hable de .profile será necesario editar el script de inicio de la shell que se esté utilizando, ya sea .profile o .cshrc o .login. Si el sistema tiene varios usuarios la mejor solución es editar los ficheros de inicio generales del sistema /etc/profile o /etc/.login.

También se debe verificar que el enlazador en modo de ejecución -runtime linker-, ld.so, puede encontrar las librerías compartidas que utiliza el sistema X Window. Esto se hace añadiendo la línea  
/usr/X11R6/lib

Al fichero /etc/ld.so.conf (si es que ya no estaba allí). A partir de este momento ya podemos configurar el sistema X Window.

Repetimos de nuevo, que si se está utilizando una distribución Linux que incluya el sistema X Window system posiblemente no sea necesaria más configuración adicional.

## **Configurando el sistema X Window**

Cuando se lanzó por primera vez el sistema X Window, era todo un desafío poderlo configurar correctamente. No obstante, con las nuevas versiones de Xfree86 y el trabajo realizado por varios distribuidores de Linux, la configuración se ha convertido en algo mucho más sencillo. El servidor X ha cambiado para soportar módulos cargables dependiendo de la tarjeta de vídeo y del chipset utilizados, haciendo más fácil su manejo. El único secreto que tiene esta configuración en los sistemas Linux actuales es conocer el comando correcto de configuración del entorno X.

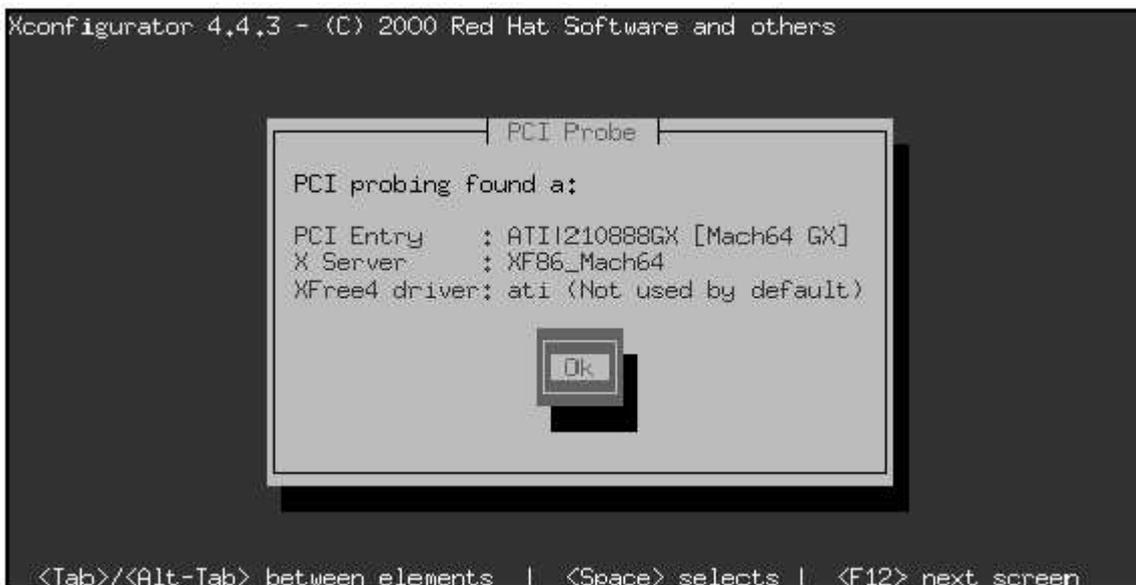
Normalmente será XF86Setup o xf86config. Otras distribuciones Linux pueden tener otros comandos diferentes: RedHat utiliza Xconfigurator, y Mandrake utiliza Xfdrake.

Todos estos comandos sirven para lo mismo, configurar el entorno X Window para que trabaje correctamente con el hardware de vídeo. Estos programas crean el fichero /etc/X11/XF86Config, que define los parámetros y el comportamiento del servidor X, incluyendo la localización de los ficheros, definiciones de fuentes, resoluciones de pantalla y configuraciones del

monitor.

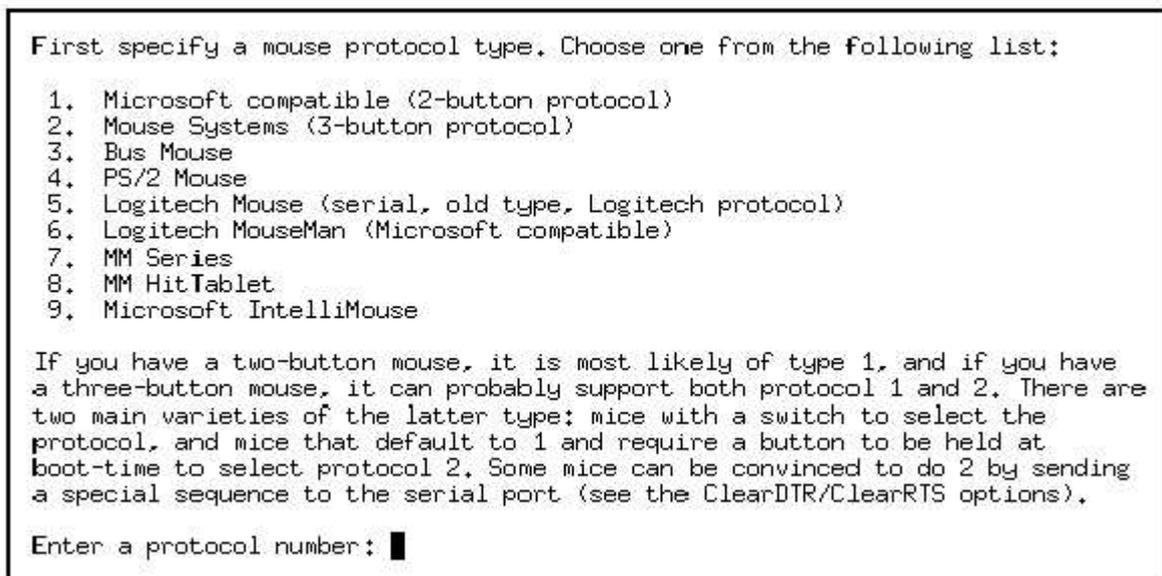
El programa XF86Setup arranca el servidor X VGA de 16 colores e interactúa con el usuario para definir los parámetros de configuración y chequear el servidor X seleccionado. RedHat y Debian Linux utilizan un programa llamado xf86config o Xconfigurator, que no inicia el servidor X hasta que no están listos para chequear la configuración.

La siguiente figura muestra al configurador Xconfigurator de RedHat confirmando el hardware de vídeo detectado:



Pantalla principal de Xconfigurator

El programa Xconfigurator es una versión mejorada de xf86config, que está totalmente basado en texto como se puede ver en la siguiente figura:



Configuración del ratón en xf86config

## Tema 110 El sistema X-Window

Todos los programas de configuración de X Window van pasando por una serie de pasos para verificar el hardware de vídeo, el ratón, el tipo de monitor, las resoluciones deseadas y la profundidad del color como se ilustra en la siguiente figura:



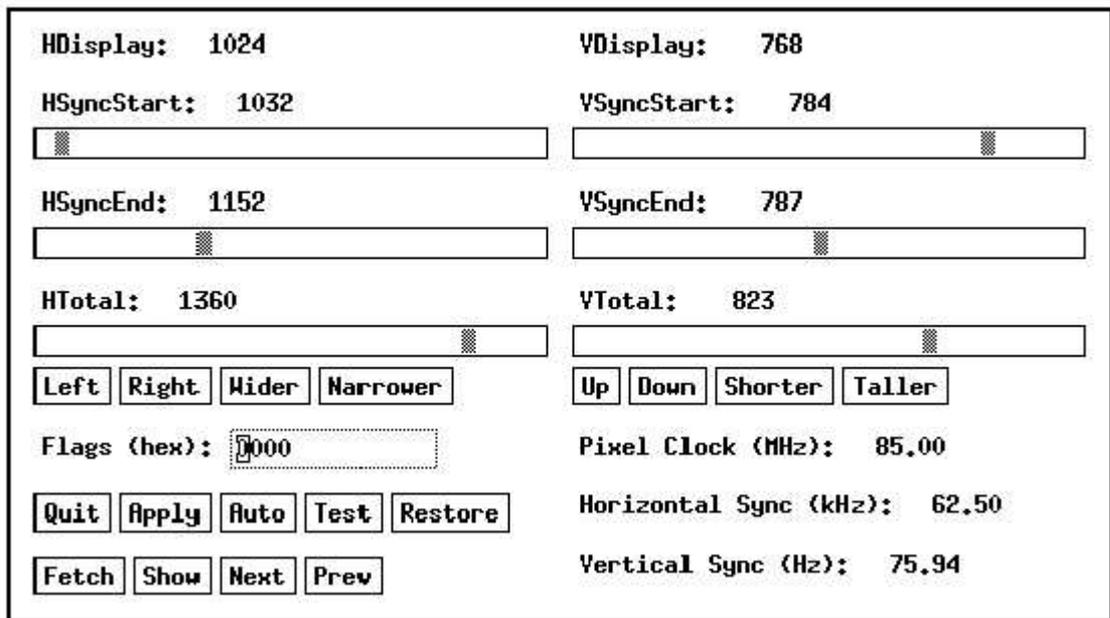
Ejemplo de selección de monitor

Con esta información, la herramienta de configuración intenta determinar que servidores X son los apropiados y, a continuación, los somete a un test para averiguar cual de ellos es el que funciona. El usuario debe decidir si la visualización es aceptable antes de que finalice el test o el programa decidirá que no es aceptable.

La documentación de X Window identifica un problema bastante molesto con los tipos de teclado, el programa XF86Setup considera que el teclado genérico es el de 102 teclas y no el típico de 101 teclas. Si se elige el de 101 teclas podríamos encontrarnos con problemas con el teclado numérico y otras teclas.

La selección de monitor es fundamental para que el servidor X conozca que prestaciones y que límites existen para la resolución y las frecuencias de refresco. Si no estamos seguros del tipo del monitor o no apareciese en el listado, siempre será mejor intentar seleccionar los tipos genéricos y luego continuar con otros de nuestro mismo fabricante, posiblemente haya que contactar con el fabricante para conocer las características del monitor y, así, poder seleccionar el tipo adecuado.

Algunos de los programas de configuración X llaman a xvidtune, que nos permite trucar los ajustes para obtener la mejor visualización. Por eso aparece un banner de aviso diciendo que podríamos dañar el monitor. Esto podría suceder aunque, como se comentó antes los monitores multisync modernos son bastante resistentes.



Pantalla de xvidtune

El programa XF86Config asume que el ratón se encuentra en /dev/mouse. Esto debería ser lo correcto ya que es lo que normalmente se define en tiempo de instalación, de todas formas, si esto no funcionase habría que averiguar a que puerto está conectado el ratón y hacer un link entre ese puerto y /dev/mouse.

### **El fichero XF86Config**

Todos los programas de configuración X terminan creando el fichero XF86Config que, normalmente, se guarda en /etc/X11.

Durante la inicialización, el servidor mira la configuración que hay en /etc/X11/XF86Config antes de presentar el interfaz X al usuario.

El servidor X busca el fichero XF86Config en los siguientes directorios:

Como usuario normal:

- /etc/X11/<cmdline>
- /usr/X11R6/etc/X11/<cmdline>
- /etc/X11/\$XF86CONFIG
- /usr/X11R6/etc/X11/\$XF86CONFIG
- /etc/X11/XF86Config-4
- /etc/X11/XF86Config
- /etc/XF86Config
- /usr/X11R6/etc/X11/XF86Config.<hostname>
- /usr/X11R6/etc/X11/XF86Config-4
- /usr/X11R6/etc/X11/XF86Config
- /usr/X11R6/lib/X11/XF86Config.<hostname>
- /usr/X11R6/lib/X11/XF86Config-4

/usr/X11R6/lib/X11/XF86Config

Como usuario root:

```
<cmdline>
/etc/X11/<cmdline>
/usr/X11R6/etc/X11/<cmdline>
$XF86CONFIG
/etc/X11/$XF86CONFIG
/usr/X11R6/etc/X11/$XF86CONFIG
$HOME/XF86Config
/etc/X11/XF86Config-4
/etc/X11/XF86Config
/etc/XF86Config
/usr/X11R6/etc/X11/XF86Config.<hostname>
/usr/X11R6/etc/X11/XF86Config-4
/usr/X11R6/etc/X11/XF86Config
/usr/X11R6/lib/X11/XF86Config.<hostname>
/usr/X11R6/lib/X11/XF86Config-4
/usr/X11R6/lib/X11/XF86Config
```

El fichero XF86Config está dividido en secciones según la información específica de cada una:

#### Secciones del fichero XF86Config

Sección	Descripción
Files	Localización de los ficheros.
Server Flags	Banderas o switches del servidor.
Module	Carga dinámica de módulos.
InputDevice	Descripción de los dispositivos de entrada.
Device	Descripción de los dispositivos gráficos.
VideoAdaptor	Descripción del adaptador de vídeo Xv.
Monitor	Descripción del monitor.
Modes	Descripción de los modos de vídeo.
Screen	Configuración de la pantalla.
ServerLayout	Presentación general.
DRI	Configuración específica de DRI.
Vendor	Configuración específica del proveedor/fabricante.

## Tema 110 El sistema X-Window

Los nombres de sección obsoletos *Keyboard* y *Pointer* aún se reconocen pero solo por motivos de compatibilidad, en su lugar debe utilizarse la sección *InputDevice*.

Las secciones *ServerLayout* enlazan los dispositivos de entrada y de salida utilizados en una sesión. Los dispositivos de entrada se describen en las secciones *InputDevice*.

Los dispositivos de salida, normalmente, están formados por varios componentes independientes (como una tarjeta gráfica y un monitor). Las tarjetas gráficas se describen en las secciones *Device* y los monitores en *Monitor*. El monitor y la tarjeta de vídeo se tratan como uno solo en las secciones *Screen*, y a esos mismos conjuntos se hace referencia en la sección *ServerLayout*.

Se puede encontrar una documentación mas extensa sobre el formato y contenido del fichero *XF86Config* en las páginas *man* de Linux.

No debe hacerse ningún cambio en el fichero *XF86Config* sin consultar en las páginas del manual la información específica sobre la sección que se va a modificar. El fichero *XF86Config* es bastante grande. (No es raro encontrar más de 400 líneas.)

El servidor de fuentes X proporciona al servidor X el trazado de las fuentes. Aunque normalmente esto siempre ha estado unido a la aplicación del servidor X, desde RedHat 6.0 se ha independizado del servidor y rueda por separado. El fichero *XF86config* tiene una sección que identifica la localización de las fuentes en el sistema.

Consideremos el siguiente ejemplo de un fichero *XF86config* file:

```
# Multiple FontPath entries are allowed (they are
#concatenated together)
# By default, Red Hat 6.0 and later now use a font
#server independent of
# the X server to render fonts.
FontPath "/usr/X11R6/lib/X11/fonts/TrueType"
FontPath "unix/:7100"
```

Esto indica que la ruta de las fuentes corresponde al directorio `/usr/X11R6/lib/X11/fonts/TrueType`, y se especifica una conexión con un servidor de fuentes. Las especificaciones del servidor de fuentes utilizan la sintaxis:

```
<trans>/<hostname>:<port-number>
```

donde `<trans>` es el tipo de transporte utilizado para conectar con el servidor de fuentes (como *unix* para Unix-domain sockets o *tcp* para una conexión TCP/IP), `<hostname>` es el nombre de la máquina que rueda el servidor de fuentes, y `<port-number>` es el número de puerto donde el servidor de fuentes escucha las peticiones (normalmente 7100).

Teniendo esto en mente, miremos de nuevo al ejemplo:

```
FontPath "unix/:7100"
```

Sabemos que la ruta de fuentes apunta a un transporte del tipo Unix-domain socket. La ausencia de un nombre de máquina a la izquierda de los dos puntos implica que la conexión es con la máquina local en el puerto 7100. Si el servidor de fuentes que queremos utilizar estuviese en otra

máquina diferente, la entrada anterior podría ser:

```
FontPath "unix/fontserver.mydomain.com:7100"
```

Cuando no se especifica la entrada *FontPath* en el fichero XF86Config, el servidor retorna al modo histórico de fuentes precompiladas:

```
/usr/X11R6/lib/X11/fonts/misc/  
/usr/X11R6/lib/X11/fonts/Speedo/  
/usr/X11R6/lib/X11/fonts/Type1/  
/usr/X11R6/lib/X11/fonts/CID/  
/usr/X11R6/lib/X11/fonts/75dpi/  
/usr/X11R6/lib/X11/fonts/100dpi/
```

La lista *FontPath* recomendada consta de los siguientes elementos:

```
/usr/X11R6/lib/X11/fonts/local/  
/usr/X11R6/lib/X11/fonts/misc/  
/usr/X11R6/lib/X11/fonts/75dpi:unscaled  
/usr/X11R6/lib/X11/fonts/100dpi:unscaled  
/usr/X11R6/lib/X11/fonts/Type1/  
/usr/X11R6/lib/X11/fonts/CID/  
/usr/X11R6/lib/X11/fonts/Speedo/  
/usr/X11R6/lib/X11/fonts/75dpi/  
/usr/X11R6/lib/X11/fonts/100dpi/
```

Las rutas de fuentes que se detecte que sean incorrectas se eliminarán de la lista de rutas durante la inicialización del servidor. Si hubiese directorios adicionales de fuentes instaladas deberían añadirse a la lista mediante entradas *FontPath*. No obstante, el fichero XF86Config es utilizado por el servidor X y no por el servidor de fuentes X. El servidor de fuentes X, conocido normalmente como XFS, se arranca durante la inicialización del sistema. XFS lee su propio fichero de configuración que define su modo de operación independientemente del servidor X. Este fichero se encuentra en `/etc/X11/fs/config`. A continuación tenemos un ejemplo de un fichero de configuración XFS:

```
#  
# Default font server configuration file for Red Hat Linux  
#  
# allow a max of 10 clients to connect to this font server  
client-limit = 10  
# when a font server reaches its limit, start up a new one  
clone-self = on  
# alternate font servers for clients to use  
#alternate-servers = foo:7101,bar:7102  
# where to look for fonts  
#  
catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,  
/usr/X11R6/lib/X11/fonts/75dpi:unscaled,  
/usr/X11R6/lib/X11/fonts/misc,  
/usr/X11R6/lib/X11/fonts/Type1,  
/usr/X11R6/lib/X11/fonts/Speedo,
```

```
/usr/X11R6/lib/X11/fonts/75dpi,  
/usr/share/fonts/default/Type1  
# in 12 points, decipoints  
default-point-size = 120  
# 100 x 100 and 75 x 75  
default-resolutions = 75,75,100,100  
# use lazy loading on 16 bit (usually Asian) fonts  
deferglyphs = 16  
# how to log errors  
use-syslog = on  
# don't listen to TCP ports by default for security reasons  
no-listen = tcp
```

Particularmente interesante es la palabra clave *catalogue*, ya que es esencialmente equivalente a la palabra clave *FontPath* de CF86config. Por tanto, si se hacen cambios en la directiva *FontPath* de XF86config, también habría que hacer los mismos cambios en *catalogue* de */etc/X11/xfs/config* para que el servidor de fuentes se entere.

El servidor de fuentes X se inicia normalmente durante la inicialización del sistema en */etc/rc.d/init.d/xfs*. Este script puede ser utilizado además para parar y/o reiniciar el servidor de fuentes X.

Se puede encontrar más información sobre el servidor de fuentes X en las páginas man correspondientes a XFS.

## Controlando X

El servidor X puede controlarse para proporcionar determinadas configuraciones cuando arranquen los clientes utilizando el fichero *.Xresources*. Si existiese este archivo, debería encontrarse en el directorio *home* del usuario. El fichero *.Xresources* no se crea automáticamente y hay una configuración por defecto válida para todo el sistema. Este fichero contiene directivas que se aplicarán al cliente X cuando se inicialice. El siguiente extracto de un fichero *.Xresources* indica como se mostrara el color en un xterm

```
xterm_color*background: Black  
xterm_color*foreground: Wheat  
xterm_color*cursorColor: Orchid  
xterm_color*reverseVideo: false  
xterm_color*scrollBar: true  
xterm_color*saveLines: 5000  
xterm_color*reverseWrap: true  
xterm_color*font: fixed  
xterm_color.geometry: 80x25+20+20  
xterm_color*fullCursor: true  
xterm_color*scrollTtyOutput: off  
xterm_color*scrollKey: on  
xterm_color*VT100.Translations: #override\n\  
<KeyPress>Prior : scroll-back(1,page)\n\  
<KeyPress>Next : scroll-forw(1,page)  
xterm_color*titleBar: false
```

Cada una de estas directivas es una directiva del sistema X Window que describe como se visualizará el cliente. Cada línea consiste de un nombre de cliente seguido por un asterisco y del parámetro X.

A través de un fichero `.Xresources` cuidadosamente ensamblado, el usuario puede manipular y definir el aspecto que presentará cada aplicación al iniciarse.

## Iniciando X

Hay dos métodos principales para iniciar una sesión X. Uno es conectando en un entorno de texto que, después, arranque el servidor X. El otro es utilizando el sistema de login gráfico de XDM. En esta sección se describirá como hacerlo desde una shell de texto.

Para tener acceso al servidor X y a los clientes locales relacionados, `/usr/X11R6/bin` ha de encontrarse en la variable `PATH`, si no estuviese habría que añadirlo en el `.profile`.

La forma más fácil de iniciar el sistema X Window es a través del comando **startx**. **startx** a su vez, llama a los programas necesarios para inicializar el entorno e inicializar el servidor X. Una vez arrancado el servidor X, se ejecuta el fichero `.xinitrc` situado en el directorio *home* del usuario. Si no existiese se ejecutaría por defecto el fichero `/usr/X11R6/lib/X11/xinit/xinitrc`. Por ejemplo, el fichero `xinitrc` podría contener lo siguiente:

```
#!/bin/sh

xterm -fn 7x13bold -geometry 80x32+10+50 &
xterm -fn 9x15bold -geometry 80x34+30-10 &
oclock -geometry 70x70-7+7 &
xsetroot -solid midnightblue &

exec twm
```

Este script arranca dos clientes `xterm` y un `oclock`, y selecciona el color de fondo de la ventana de `root` como *midnight blue*. Después arranca `twm`, el manejador de ventanas.

Hay que tener en cuenta que `twm` se ejecuta con el comando shell `exec`, lo que provoca que el proceso `xinit` sea reemplazado por `twm`. Una vez finalice el proceso `twm`, el servidor X se cerrará. El comando **twm** es un gestor de ventanas, hay muchos gestores disponibles como `twm`, `kwm`, y `fvwm95`, por nombrar solo unos pocos.

El último comando en `.xinitrc` debe ser arrancado con **exec** y no debe ser enviado al `background`. Si no se utilizase **exec** o se enviase el comando al `background`, el servidor X finalizaría justo después de completar la ejecución de los clientes del fichero `xinitrc`.

# **Tema 110.2**

# **Configurando el**

# **inicio de sesiones**

# **gráficas**



## **Introducción**

En este capítulo se verá como configurar el gestor de sesiones gráficas, arrancar en modo gráfico, usar kdm, gdm, xdm ...

Este tema tiene un peso (importancia) de 3 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

## Configurando el proceso de inicio

Después de ejecutarse LILO, el kernel ejecuta su primer proceso: init. Es importante resaltar que al ser el primer proceso, Init siempre tendrá el pid=1. Todos los demás procesos iniciados después de Init, son procesos hijos suyos. Eso puede verse reflejado en el siguiente fragmento:

```
[invitado@iMac invitado]$ ps -eaf
UID      PID  PPID  C STIME TTY      TIME CMD
root      1    0  0 Feb24 ?        00:00:00 init [5]
root      2    1  0 Feb24 ?        00:00:00 [ksoftirqd/0]
root      3    1  0 Feb24 ?        00:00:00 [events/0]
root      4    3  0 Feb24 ?        00:00:00 [khelper]
root     18    3  0 Feb24 ?        00:00:00 [kblockd/0]
```

Cuando se ejecuta Init, este obtiene su configuración del fichero /etc/inittab. Cada una de las entradas en el fichero /etc/inittab tiene la siguiente sintaxis:

**id:runlevels:acción:proceso**

En cualquiera de las líneas en las que aparezca el símbolo #, se ignorará todo su contenido a la derecha de este. La siguiente tabla describe los campos:

<i>Campo</i>	<i>Descripción</i>
id	Una cadena única de uno a cuatro caracteres que identifica una entrada en el fichero inittab.
runlevels	La lista de runlevels, o niveles de ejecución para los que la acción de esta entrada va a llevarse a cabo.
acción	Especifica la acción que se llevará a cabo.
proceso	El proceso que deberá ejecutarse.

La siguiente tabla describe las acciones a tomar

<i>Acción</i>	<i>Descripción</i>
Respawn	El proceso se regenerará en cuanto sea terminado. Lo podréis ver con el proceso de login getty, que necesita ser regenerado siempre.
Wait	El proceso se iniciará una sola vez e Init esperará a que termine antes de continuar.
Once	El proceso se iniciará una sola vez, cuando corresponda, en su runlevel.
Boot	El proceso se iniciará durante el arranque y se ignorará cualquier entrada de runlevel que pudiera tener.
Bootwait	El proceso se iniciará durante el arranque e Init esperará a que termine antes de continuar.
Initdefault	Especifica el runlevel que se ejecutará por defecto durante el arranque.
Sysinit	Esta entrada se ejecutará durante el arranque, antes que cualquier otra entrada boot o bootwait. Init ignorará el campo runlevel para este tipo de entradas.

<i>Acción</i>	<i>Descripción</i>
Powerwait	Esta entrada se procesa cuando falla la toma de energía del sistema. Por ejemplo cuando lo indica un SAI. Init esperará a que termine antes de continuar.
Powerfail	Esta entrada se procesa cuando falla la toma de energía del sistema, con la diferencia que Init no esperará a que termine para continuar.
Powerokwait	Esta entrada se ejecuta cuando la toma de energía vuelve a estar disponible e Init esperará a que termine antes de continuar.
Powerfailnow	Esta entrada se ejecuta cuando el SAI indica al sistema que sus baterías están a punto de agotarse.
Ctrlaltdel	Esta entrada se procesa cuando un usuario pulsa la combinación de teclas Ctrl-Alt-Supr.
Kbrequest	Ejecuta el proceso indicado cuando se pulsa la combinación especial de teclas definida.

Veamos un ejemplo de un fichero /etc/inittab:

```
id:3:initdefault
```

Initdefault establece el runlevel por defecto que el sistema usará para arrancar, en este caso el runlevel 3.

```
# System initialization
si::sysinit:/etc/rc.d/rc.sysinit
```

Esta línea ejecuta el script /etc/rc.d/rc.sysinit durante la inicialización del sistema. Este script va a ejecutarse antes que ningún otro, e init esperará a que acabe antes de proseguir.

```
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

Estas líneas configuran los scripts a ejecutar bajo cada runlevel. Si por ejemplo el sistema cambia a runlevel 5, se ejecutará “/etc/rc.d/rc 5”. En este caso el tercer campo a “wait” especifica que init también esperará a que finalice la ejecución antes de proseguir.

```
# Things to run in every runlevel
ud::once:/sbin/update
```

Esta línea establece que /sbin/update deberá ser ejecutado en cualquier runlevel.

```
# Trap CTRL-Alt-Del
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Esta otra línea establece que cuando se pulse la combinación Ctrl-Alt-Del, deberá ejecutarse el comando “/sbin/shutdown -t3 -t now”.

```
pf::powerfail:/sbin/shutdown -f -h +2 “Power failure: System shutting down”
```

Esta línea establece que si ocurre un fallo en la toma de energía, debe apagarse el sistema.

```
pr:12345:powerokwait:/sbin/shutdown -c “Power restored: Shutdown cancelled”
```

Esta línea aborta el cierre del sistema si este detecta que la energía ha sido restablecida.

```
1:2345:respawn:/sbin/mingetty tty1  
2:2345:respawn:/sbin/mingetty tty2  
3:2345:respawn:/sbin/mingetty tty3  
4:2345:respawn:/sbin/mingetty tty4  
5:2345:respawn:/sbin/mingetty tty5  
6:2345:respawn:/sbin/mingetty tty6
```

El proceso “/sbin/mingetty” se ejecutará para los runlevels 2,3,4 y 5. mingetty establece el login de las consolas. “Respawn” indica que se deberá regenerar en caso de que finalice su ejecución.

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Esta última línea inicia el sistema gráfico para el runlevel 5.

### **Los arranques gráficos: xdm**

Dentro de los diversos tipos de arranques en modo gráfico, xdm (que viene de serie con Xfree86), es el más sencillo. xdm presenta una GUI sencilla para registrar el login en el sistema, cuando nos identificamos correctamente, aparecemos directamente en nuestra sesión X Window. Si salimos de la sesión, el sistema nos devolverá a xdm.

### **Activando y desactivando XDM**

Cuando está activado, xdm se inicia durante el arranque a través de un runlevel. En Debian por ejemplo, se activa usando un enlace a un script llamado S99xdm, mientras que en Red Hat, el sistema cambia de runlevel 3 al 5.

### **Configurando XDM**

Xdm puede ser difícil y complicado de configurar, puesto que tiene varios ficheros que deben ser mantenidos. Las secciones siguientes cubren la configuración de los ficheros más importantes para configurar xdm.

#### **xdm-config**

Este es el fichero de configuración principal de xdm. Este fichero establece la operativa básica de xdm, que para la mayoría de instalaciones no deberá ser cambiado nunca, pero si es necesaria la configuración del sistema para permitir logins remotos, deberá ser necesario realizar algún ajuste. No entraremos en mucho detalle, pero por ejemplo, si deseáramos permitir logins remotos a través de xdm, deberíamos cambiar la siguiente línea:

```
DisplayManager.requestPort: 0
```

Y comentarla para permitir a xdm escuchar consultas:

```
!DisplayManager.requestPort: 0
```

### **Xsession**

Xdm usa el script Xsession para varias cosas:

- Recoger los errores durante la inicialización de la sesión y redirigirlo al fichero `${HOME}/.xsession-errors`.
- Cargar variables.
- Cargar los mapas de teclado correspondientes.
- Y averiguar que escritorio debe proporcionar a cada usuario para su sesión (KDE, Gnome ...)

### **Xresources**

En Xresources se definen los recursos que permiten personalizar la pantalla de login para xdm. Se usa el formato estándar Xresource, discutido más adelante, para cambiar colores, tipos de fuente y demás.

## **KDM y GDM**

Kdm no es más que la versión del KDE del xdm, y el GDM la versión de Gnome. Con un aspecto mucho más mejorado y con más opciones.

Para escoger que gestor de arranque utilizar en el arranque:

Red Hat: hay que editar el fichero `/etc/inittab`, buscar la cadena `xdm` y reemplazarla con `kdm` o `gdm`.

Debian: Editar el fichero `/etc/X11/default-display-manager` y poner el gestor deseado.

# **Tema 110.4**

# **Configurando el**

# **administrador**

# **gráfico**

## **Introducción**

En este capítulo se verá como configurar el administrador gráfico de las X. Se verá como arrancar diferentes entornos gráficos, terminales, etc.

Nota: Este tema puede tener contenidos que a fecha de hoy estén desfasados o cambiados.

Este tema tiene un peso (importancia) de 5 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

## Arrancando X

Hay dos formas de arrancar X en el sistema. El primer método consiste en hacerlo manualmente desde la línea de comandos tras el login. La segunda forma es hacer que el sistema arranque en modo gráfico con un login GUI.

## Arranque manual de X

Se puede arrancar manualmente el sistema X Window con el comando *startx*. A continuación seguiremos el flujo de scripts que comienzan tras la ejecución de *startx*. El problema es que las distintas distribuciones no siguen el mismo camino a través de los scripts de inicio de X. Es muy recomendable recorrer los scripts de la distribución y versión que tenemos instalada para ver exactamente lo que hacen. Además es una experiencia excelente.

```
$startx
```

Este comando suele ser un script que inicializa una serie de variables y, finalmente, llama a la aplicación *xinit*. A continuación se incluye un ejemplo de un script *startx* con comentarios al final:

```
#!/bin/sh
#
# (c) 1999 Red Hat Software, Inc.
bindir=/usr/X11R6/bin
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/etc/X11/xinit/xinitrc
sysserverrc=/etc/X11/xinit/xserverrc
clientargs=""
serverargs=""
```

Esta parte inicializa algunas variables que apuntan a otros ficheros de comandos que se invocarán más adelante.

```
if [ -f $userclientrc ]; then
  clientargs=$userclientrc
else if [ -f $sysclientrc ]; then
  clientargs=$sysclientrc
fi
fi

if [ -f $userserverrc ]; then
  server=$userserverrc
else if [ -f $sysserverrc ]; then
  server=$sysserverrc
fi
fi
```

Estas líneas hacen un *test -f* para comprobar si existen los ficheros indicados en las variables

anteriores.

La siguiente sección, que aquí omitiremos, ejecuta los ficheros de comandos indicados en las variables, los ficheros *xinitrc* y *xserverrc*.

```
# set up default Xauth info for this machine
mcookie=`mcookie`
serverargs="$serverargs -auth $HOME/.Xauthority"
xauth add $display . $mcookie
xauth add `hostname` -f $display . $mcookie
```

Este bloque configura la información de seguridad de X ejecutando la utilidad *xauth*.

```
xinit $clientargs -- $server $display $serverargs
```

Finalmente, el script *startx* ejecuta *xinit*.

Consejo para el examen: El comando *startx* es un script que ejecuta el proceso *xinit*.

### **Los ficheros *xinitrc* y *.xinitrc***

Cada vez que arranca el proceso *xinit* se ejecuta un fichero de comandos. Cada usuario puede crear su fichero personal *.xinitrc* en su directorio home. Si este fichero no existiese se ejecutaría en su lugar el fichero genérico *xinitrc* que se encuentra en */etc/X11/xinit* o en */usr/X11/xinit/xinitrc*. Si se incluyen varias tareas dentro del fichero *.xinitrc*, la última no debería enviarse al background porque el servidor X podría terminar.

Consejo para el examen: Cuando un usuario arranca X con el comando *startx* se ejecuta el script *.xinitrc* del usuario, si no existiese se ejecutaría el script genérico *xinitrc*.

### **Los ficheros *Xclients* y *.Xclients***

Algunas distribuciones hacen que el fichero *xinitrc* llame al script *Xclients*. Red Hat utiliza este fichero para chequear y saber que manejadores de ventanas están instalados y para arrancar aquel que el usuario ha configurado como favorito. Un usuario puede tener su propio fichero *Xclients* en *~/.Xclients*.

### **Utilizando X**

Tras librar la batalla de la instalación y configuración de X nos encontramos con la recompensa de poder disfrutarlo. El entorno X es altamente personalizable hasta el punto que sorprenderá a algunos usuarios nuevos de Linux y X.

### Eligiendo un entorno o manejador de ventanas

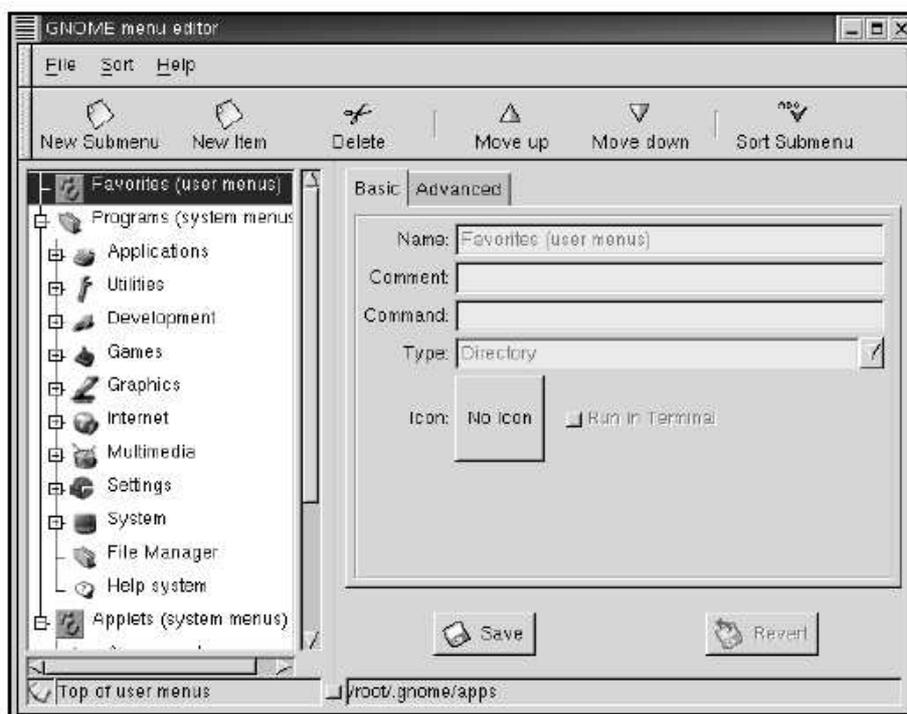
Hay disponible una amplia variedad de manejadores de ventanas, unos más conocidos que otros. Muchas personas utilizan solo un manejador de ventanas determinado mientras otras utilizan un entorno completo de escritorio como como KDE o GNOME, que combina un manejador de ventanas junto con una serie de aplicaciones y herramientas integradas en el mismo.

El manejador de ventanas escogido se arranca normalmente desde el fichero Xclient, Xsession, o .xinitrc, dependiendo de la distribución y de la elección del usuario. Red Hat utiliza el fichero Xclient mientras que Debian usa un fichero Xsession genérico para los usuarios locales y remotos. Para ejecutar un manejador de ventanas distinto a los incorporados en esos ficheros se puede especificar manualmente en el fichero .xinitrc dentro del directorio home. Este script se ejecutará cada vez que se arranque el proceso *xinit*.

Una vez escogido el entorno X podemos trabajar para personalizar el sistema. Un método podría ser la utilización de temas que cambiarán el aspecto general del sistema. <http://www.themes.org> es un sitio web muy popular dedicado a temas.

Aunque los manejadores de ventanas necesitaban modificar a mano los ficheros de configuración para hacer hasta el cambio más simple, los manejadores de ventanas actuales suelen incluir un panel de control con herramientas para realizar estas tareas más fácilmente.

A continuación se muestra una imagen del editor de menús de GNOME.



### Utilizando los clientes X

Hay varios parámetros de línea de comandos utilizados por la mayor parte de las aplicaciones clientes X. Estos parámetros nos permiten establecer la configuración de inicio mediante una sintaxis estandar. Son conocidos como “X Toolkit options” (*Opciones de la colección de herramientas de X*), y aunque puedan ser un poco pesados para teclearlos a mano si que son muy prácticos para utilizarlos en un script o en un menú

La siguiente tabla muestra estos parámetros.

<i>Opción</i>	<i>Función</i>
-bg <color> o -background <color>	Establece el color de fondo por defecto para la aplicación.
-fg <color> o -foreground <color>	Establece el color de texto por defecto para la aplicación.
-bd <color> o -bordercolor <color>	Establece el color del borde de ventana por defecto para la aplicación.
-bw <número> o -borderwidth <número >	Establece la anchura en pixels del borde de la ventana.
-display nombrehost:display.pantalla	Especifica el nombre del host y los números de display y pantalla en los que se mostrará la aplicación.
-fn <fuente> o -font <fuente>	Especifica la fuente utilizada para el texto.
-geometry <anchura>x<altura>+<x>+<y>	Especifica el tamaño y posición inicial de la ventana.
-iconic	Le indica a la aplicación que se inicie en modo icono si fuese posible.
-name <nombre>	Especifica el nombre bajo el cual se encontrarán los recursos de las aplicaciones.
-rv o -reverse	Le indica a la aplicación que simule vídeo inverso si fuese posible.
+rv	Le indica a la aplicación que no intente simular vídeo inverso.

### **Utilizando un emulador de terminal**

Muchos usuarios ejecutan X como una interfaz gráfica de múltiples emuladores de terminal simultáneos en una sola pantalla. Existen diferentes emuladores de terminal como para satisfacer las necesidades de todo el mundo. A continuación se indican varios de los más populares.

xterm

El emulador de terminal “estándar” xterm existe desde hace mucho tiempo. Proporciona emulaciones DEC VT102/VT220 y Tektronix 4104 para aplicaciones . Soporta un gran número de opciones de línea de comandos para configurar el emulador en tiempo de ejecución y también soporta las opciones estándar X Toolkit que se mostraron anteriormente

rxvt

Para aquellos usuarios que no necesiten toda la funcionalidad de xterm existe rxvt. rxvt no emula tantos sistemas como hace xterm y tampoco incluye soporte de las opciones X Toolkit. El beneficio es una menor utilización de memoria puesto que fue diseñado para ser utilizado en sistemas con muchos emuladores de terminal abiertos simultáneamente.

aterm

aterm es similar a rxvt en tanto que no soporta tantas emulaciones como xterm. Tampoco soporta

las opciones X Toolkit. Fue diseñado para trabajar con el manejador de ventanas AfterStep, pero no es indispensable. aterm ofrece varias opciones que no incluye xterm, entre ellas un modo de transparencia en el cual se ve el fondo del escritorio a través de la ventana.

#### Eterm

Si buscamos el emulador de terminal de mejor aspecto y más configurable, Eterm es ese. Fue diseñado para trabajar con el manejador de ventanas Enlightenment, y se nota. Necesita más memoria que otros emuladores pero admite tantas configuraciones y personalizaciones como se pueda desear. También soporta temas.

#### gnome-terminal

El entorno de escritorio GNOME viene con su propio emulador de terminal GNOME-compatible. Está escrito utilizando las librerías GTK (GIMP Tool Kit) así que tendrá el mismo aspecto y funcionamiento del tema GNOME que se tenga instalado. Es un emulador de terminal muy completo y una buena elección si se utiliza GNOME.

#### konsole

Para no ser menos, el grupo KDE incluye en su entorno de escritorio un emulador de terminal llamado konsole. Se trata de un emulador completo que hereda el aspecto y funcionamiento de KDE.

## **Personalizando las aplicaciones X**

Como ya se ha comentado, el entorno X es altamente personalizable. Las aplicaciones escritas para utilizar entornos de escritorio como GNOME y KDE heredan los aspectos y funcionalidades que hayan sido configurados en esos entornos, ¿Pero que pasa con el resto de las aplicaciones?. Hay muchas aplicaciones que no han sido escritas para un entorno de escritorio y solamente utilizan las librerías estándar X Toolkit, pero aun así nos proporcionan métodos para personalizar su apariencia.

Las aplicaciones que utilizan las librerías X Toolkit pueden personalizarse utilizando el formato Xresource. En /usr/X11R6/lib/X11/appdefaults o en /etc/X11/app-defaults podemos encontrar numerosos ficheros de ejemplo de Xresource. El nombre de cada uno de ellos es el de la aplicación a la que se corresponden. A continuación veremos un ejemplo tomado del fichero Xman, el cual se corresponde con la aplicación *xman* que visualiza las páginas de *man*.

```
*manualBrowser.Title: Manual Page
*manualBrowser.IconName: Manual Page
*manualBrowser.geometry: 600x600
```

Aquí podemos ver el título por defecto, el nombre del icono y la geometría de la ventana. De hecho, la sintaxis de estas entradas puede ser muy compleja, pero las más simples que serán las que más probablemente tengamos que cambiar siguen el formato de éste ejemplo que es el nombre de la aplicación seguido por la propiedad a cambiar.

Al cambiar algún ajuste en un fichero de los que se encuentran en apps-default, este ajuste quedará alterado para todos los usuarios del sistema. Para cambiar el comportamiento de la aplicación solo para un usuario deberíamos poner las entradas del fichero Xresource en otro fichero

llamado .Xdefaults dentro del directorio home del usuario correspondiente.

Consejo para el examen: El fichero .Xdefaults guarda las personalizaciones del usuario para las aplicaciones X.

### Utilizando las teclas especiales

Hay varias combinaciones especiales de teclas que pueden utilizarse dentro de X. Los usuarios que solían cambiar de consolas virtuales en Linux con la combinación Alt-TecladeFunción se sorprenderán al ver que esto ya no funciona en X.

La siguiente tabla nos muestra las combinaciones especiales.

#### Combinaciones de Teclas

Combinación	Función
Ctrl-Alt-<+ del Teclado numérico>	Cambia a una resolución de mayor vídeo si estuviese configurada.
Ctrl-Alt-<- del Teclado numérico>	Cambia a una resolución menor de vídeo si estuviese configurada.
Ctrl-Alt-Retroceso	Salida rápida de X a no ser que esta opción estuviese desactivada en XF86Config.
Ctrl-Alt-<F1 hasta F6>	Cambia a las consolas de texto.
Ctrl-Alt-<F7>	Cambia de nuevo al modo gráfico desde una consola de texto.

### Gestionando aplicaciones con mal comportamiento

Debido a la naturaleza gráfica de las aplicaciones X, éstas son normalmente mas complejas y difíciles de desarrollar que las aplicaciones de texto. Por esta causa muchas aplicaciones X se han ganado la mala reputación de ser inestables o de causar problemas.

Una de las aplicaciones que más quejas ha reportado es el navegador Netscape Navigator, pero no es el único culpable.

Algunas aplicaciones, simplemente caen mientras otras no se cierran cuando se les indica, se quedan abiertas y siguen consumiendo recursos que podrían utilizarse para otras cosas.

Si detectásemos que el sistema rueda más lento de lo normal o escuchamos continuamente el disco duro trabajar mientras intercambia memoria con la partición de swap, deberíamos comprobar la lista de procesos. Asegurémonos de que ninguna aplicación que pensamos que debería estar cerrada aparece como abierta y ejecutándose en el background.

Consejo para el examen: Muchas de las aplicaciones inestables ruedan en X, los objetivos del examen mencionan específicamente *Netscape Navigator* y *tkrat* (un agente de correo gráfico), así que es conveniente recordarlas.

## Ejecutando Remotamente X y Clientes

Una característica muy buena de X es la posibilidad de mostrar transparentemente aplicaciones a través de una red. De esta forma podemos configurar equipos económicos de sobremesa para que los usuarios manejen aplicaciones que se ejecutan en un gran servidor de red. X permite tanto ejecutar remotamente aplicaciones sueltas como iniciar un escritorio completo desde un sistema remoto.

## Configurando la seguridad de X

X soporta varios mecanismos de autenticación, algunos muy complejos. Para la mayoría de los usuarios el método básico basado en host funciona bien y es fácil de configurar y manejar. La seguridad basada en el host se configura con el comando *xhost*. Se pueden consultar los ajustes de seguridad actuales ejecutando *xhost* sin opciones.

```
[root@redhat /root]# xhost
access control enabled, only authorized clients can connect
INET:brain
INET:marvin
```

También podemos utilizar el comando *xhost* para ver si disponemos de permisos para mostrar clientes en un sistema remoto. Esto se consigue cambiando el valor de la variable DISPLAY al del host remoto y después ejecutando *xhost*. Por ejemplo para ver si podemos mostrar aplicaciones en el host `redhat.the-nashes.net` escribiríamos:

```
export DISPLAY=redhat.the-nashes.net:0.0
xhost
```

En el siguiente apartado volveremos con la variable DISPLAY.

El comando *xhost* emplea solo unas pocas opciones de línea de comandos. Para habilitar la autenticación basada en host y limpiar todos los hosts permitidos utilizaremos *xhost -*. Por ejemplo:

```
[root@redhat /root]# xhost -
access control enabled, only authorized clients can connect
```

Para desconectar todas las autenticaciones y permitir a todo el mundo que se conecte a nuestro sistema y a nuestras aplicaciones escribiríamos *xhost +*. Por ejemplo:

```
[root@redhat /root]# xhost +
access control disabled, clients can connect from any host
```

¡Precaución!: Ejecutar *xhost +* no es una buena idea ya que cualquier usuario podría conectarse a nuestro servidor X.

Para autorizar a determinados hosts a conectarse utilizamos también la opción *+*. Por ejemplo, para dar permisos de conexión a tres hosts escribiríamos lo siguiente:

```
[root@redhat /root]# xhost +deedee marvin brain
deedee being added to access control list
marvin being added to access control list
brain being added to access control list
```

Para eliminar un host de la lista de permisos utilizaremos la opción -.  
Por ejemplo, para eliminar a *deedee* escribiríamos lo siguiente:

```
[root@redhat /root]# xhost -deedee
deedee being removed from access control list
```

Finalmente, para asegurarnos de que nuestra autenticación es correcta escribiríamos solamente *xhost*:

```
[root@redhat /root]# xhost
access control disabled, clients can connect from any host
INET:brain
INET:marvin
```

Consejo para el examen: La autenticación basada en host se configura con el comando *xhost*.

## Configurando Clientes Remotos

Hay dos formas básicas de indicar a una aplicación que se muestre en un sistema remoto. La primera emplea la variable de entorno DISPLAY. El formato de uso de la variable DISPLAY es el siguiente:

```
DISPLAY=nombrehost:display.pantalla
```

Por ejemplo:

```
DISPLAY=marvin.the-nashes.net:0.0
```

El campo *nombrehost* puede ser tanto un nombre DNS como una dirección IP. Si solo hubiese un usuario utilizando un servidor X en el sistema remoto el número de display sería 0.

El número de pantalla se utiliza solo en entornos multipantalla y puede omitirse si solo se utiliza una pantalla.

Después de modificar la variable DISPLAY, cualquier cliente X que se inicie se mostrará automáticamente en el sistema especificado en la variable, si la autenticación así lo permitiese.

El segundo método para mostrar clientes remotamente consiste en emplear la opción de línea de comando *-display* reconocida por la mayoría de los clientes.

La sintaxis es la siguiente:

```
xclient -display nombrehost:display.pantalla [argumentos de xclient]
```

El formato, como se puede observar, es el mismo que el empleado con la variable DISPLAY.

Éste método es práctico para cuando solo deseamos mostrar uno o dos clientes en otro sistema.

Configurando el login remoto

Otra función muy práctica de XDM es la de permitirnos conectarnos remotamente a otro sistema y utilizar nuestro sistema local solo para visualización. También podemos configurar un sistema para proporcionar un cliente con un menú de distintos sistemas de login. La funcionalidad subyacente la proporciona y controla XDMCP (X Display Manager Control Protocol).

Estos ajustes deberían funcionar con todos los clientes X y servidores con soporte XDMCP.

Referencia cruzada: El “Linux Terminal Server Project” (Proyecto de Servidor de Terminales para Linux) ha creado un sistema de login que puede ser empleado en estaciones de trabajo sin discos. Podemos encontrar más información en <http://www.ltsp.org>.

Conectándonos a sistemas remotos

Si un sistema remoto rueda XDM, es fácil conectar con dicho sistema desde X y ejecutar aplicaciones desde él.

Se utiliza la siguiente sintaxis:

**X -query <servidor nombre>**

X también soporta la posibilidad de buscar en la red local cualquier servidor que ruede XDM. Esto se consigue con el siguiente comando:

**X -broadcast**

Los sistemas Linux también puede configurarse para proporcionar una lista de hosts con los que podemos conectarnos, esta lista se llama chooser. Para solicitar un chooser emplearíamos el siguiente comando:

**X -indirect <nombre servidor>**

Consejo para el examen: Asegurémonos de conocer la diferencia entre estos tres comandos para el examen.

***Bibliografía y enlaces recomendados***

LPI 1 Certification Bible (Bible) by Angie Nash, Jason Nash  
John Wiley & Sons; Bk&CD-Rom edition (July 1, 2001) ISBN: 0764547720

LPI Linux Certification in a Nutshell by Jeffrey Dean  
O'Reilly & Associates; 1st ed edition (May 15, 2001) ISBN: 1565927486

CramSession's LPI General Linux Part 1 : Certification Study Guide  
CramSession.com; ISBN: B000079Y0V; (August 17, 2000)

Referencias Unix Reviews  
<http://www.unixreview.com/documents/s=7459/uni1038932969999/>

Página LPI: [www.lpi.org](http://www.lpi.org)

Apuntes IBM: <http://www-106.ibm.com/developerworks/edu/l-dw-linux-lpir21-i.html>

Manuales GPL: <http://www.nongnu.org/lpi-manuals/>