
Curso Básico de X-Window

Contenidos

1 Prefacio	2
2 Introducción.....	2
3 El Modelo Cliente-Servidor	2
4 Displays	3
5 Tipos de Sesión X. Inicio de sesión.....	5
5.1 Acceso a través de la consola	5
5.2 Acceso a través de un X-Terminal	5
6 Configuración de la Sesión X	5
7 El Window Manager	6
8 Clientes X y Recursos X.....	7
9 Opciones en Línea de Comandos	8
10 Entornos de escritorio	9
11 Referencias	9

Copyright © 2002 Jorge Juan-Chico <jjchico@imse.cnm.es>

Todas las Marcas Registradas son propiedad de sus respectivos propietarios.

Se permite copiar y distribuir copias literales de este documento, siempre que la nota del copyright y esta propia nota sean conservadas en todas las copias.

Se permite la copia y distribución de versiones modificadas de este documento bajo las mismas condiciones que para copias literales, siempre que el documento resultante sea distribuido bajo los términos de una nota idéntica a ésta.

1 Prefacio

Este documento pretende ser una introducción breve y práctica al entorno de ventanas X (*X-Window*). Su enfoque es general y debe servir para cualquier implementación de X, pero se hace alguna referencia a aquellas implementaciones y entornos de escritorio sobre X que se emplean en sistemas tipo UNIX de libre distribución como GNU/Linux.

2 Introducción

El sistema de ventanas X (*X Window System*) o simplemente *X* es un estándar software que permite a los programadores, por una parte, desarrollar aplicaciones gráficas y a los usuarios, por otra parte, disponer de un entorno gráfico de trabajo.

X fue desarrollado en el Massachusetts Institute of Technology (MIT) bajo el proyecto Athena y con el apoyo de Digital Equipment Corporation (DEC). Desde entonces X se ha beneficiado del trabajo de mucha gente, hasta llegar a la versión 11 de hoy día y se ha convertido en el entorno de trabajo estándar para Sistemas Operativos tipo UNIX. Una de las variantes más populares de X es la desarrollada por el proyecto XFree86, que se inició con el fin de obtener una implementación de X para ordenadores basados en microprocesadores Intel. XFree86 es de libre distribución y en la actualidad corre sobre diversas plataformas y se usa de forma estándar en la mayoría de las versiones libres de sistemas operativos tipo UNIX: FreeBSD, GNU/Linux, etc.

Los aspectos claves de X son:

- Entorno distribuido: Independencia entre “lugar de trabajo” y “lugar de proceso”
- Independencia del Hardware: X funciona (o puede funcionar) sobre cualquier S.O.
- Máxima flexibilidad y configurabilidad.
- Absoluta libertad de elección de la interface de usuario.

3 El Modelo Cliente-Servidor

La arquitectura de X se basa en el modelo cliente-servidor. Existen dos procesos diferentes que se comunican a través de una red de comunicaciones empleando un protocolo dedicado: X Protocol:

- El servidor: que se encarga de interactuar con el usuario, controla el hardware de visualización y de entrada del usuario (teclado, ratón). Hace llegar al cliente (programa de aplicación) las acciones del usuario y muestra al usuario los resultados proporcionados por el cliente.
- El cliente: es el programa de aplicación (procesador de textos, correo electrónico, etc.). Su comunicación con el usuario se realiza siempre a través del servidor.

Este modelo aporta gran flexibilidad al sistema. Un cliente y un servidor podrán comunicarse siempre que ambos “hablen” el X Protocol, independientemente del hardware en que se ejecuten e incluso del sistema operativo sobre el que corran cliente y servidor. Además, como la comunicación se produce a través de una red de comunicaciones, cliente y servidor pueden residir en diferentes salas de un mismo centro de investigación, en diferentes edificios de un

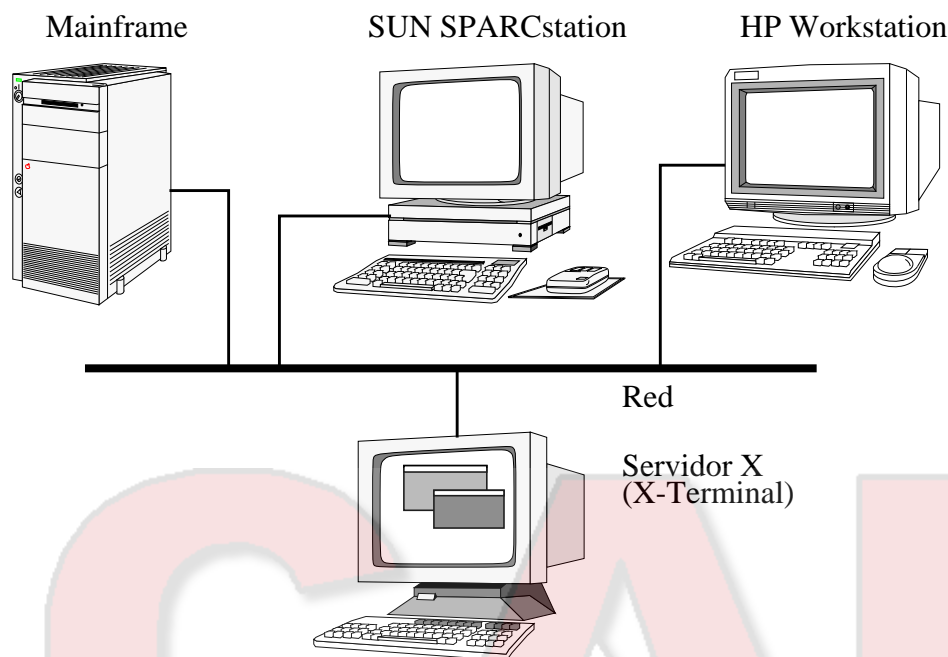


Figura 1: El modelo cliente-servidor

campus e incluso en diferentes países. Múltiples clientes pueden comunicarse con un mismo servidor. En este caso la pantalla del servidor mostrará diversas ventanas correspondientes a los diferentes clientes con los que puede interactuar el usuario.

Otro aspecto que dota a X de gran flexibilidad es el *manejo de ventanas*. A diferencia de otros entornos gráficos como Microsoft Windows o el del Apple Macintosh, X no define un tipo particular de interface de usuario, sino que proporciona una serie de primitivas para dar soporte a cualquier interface de usuario. De esta forma, las tareas de decoración de las ventanas, cambios de tamaño y posición, menús, etc. corresponden a un programa, el *Window Manager*, que se ejecuta como un cliente más.

Bajo este modelo caben diferentes configuraciones de un sistema X. En el caso más sencillo, un único ordenador con capacidades gráficas, puede correr tanto el servidor X como los diferentes clientes que precise el usuario, de forma local. En este caso no se necesita disponer de una conexión a red, pero si se tuviera, se podrían ejecutar, así mismo, clientes X sobre otras máquinas conectadas a la red de forma completamente transparente para el usuario.

Existen máquinas diseñadas expresamente para ejecutar sólo el Servidor X (X Terminals). En este caso, dichas máquinas se limitan a interactuar con el usuario y a ejecutar clientes para él de forma remota. El uso de X Terminals está muy extendido en laboratorios y centros de investigación.

4 Displays

Desde la perspectiva del usuario, todo servidor X tiene un nombre de display de la forma:

hostname:displaynumber.screennumber

Esta información es usada por la aplicación para determinar como debe conectar con el servidor

y que *screen* debe usar por defecto (en el caso de *diplays* con múltiples monitores). Los diferentes componentes del nombre del display son:

- *hostname*: El nombre de la máquina a la que el display está directamente conectado. Si se omite el hostname, se emplea el método más eficiente de comunicación con un servidor X en la propia máquina.
- *displaynumber*: El término “display” suele referirse a un conjunto de monitores que comparten un mismo teclado y puntero (ej. ratón). La mayoría de las estaciones de trabajo suelen tener un único teclado, y por tanto, un único display. Algunos sistemas multiusuario grandes pueden tener, sin embargo, varios displays. Para evitar confusión, a cada display se le asigna un número (empezando por 0) cuando se inicializa el servidor X para ese display. El *displaynumber* debe aparecer siempre en el nombre del display.
- *screennumber*: Algunos displays consisten en un único teclado y puntero pero con más de un monitor. En este caso, a cada monitor se le asigna un número de screen (*screennumber*). Si no se facilita el *screennumber*, se usa 0.

En la mayoría de los sistemas (aquellos que cumplen el estándar POSIX), el nombre del display por defecto es almacenado en una variable de entorno llamada DISPLAY. Esta variable se inicializa automáticamente en los emuladores de terminal (**xterm**). Sin embargo, a veces, cuando se hace login en otras máquinas de la red, es necesario definir DISPLAY manualmente para que las aplicaciones que corramos en la otra máquina se visualicen en nuestro display. Por ejemplo:

```
$ DISPLAY=myws:0; export DISPLAY
```

o bien, si se usa el C-Shell o una de sus variantes:

```
% setenv DISPLAY myws:0
```

La mayoría de los programas X aceptan una opción en línea de comandos para sustituir temporalmente el valor de DISPLAY. Se suele usar para abrir ventanas en el screen de otra persona o bien como parte de una conexión remota para abrir un **xterm** que apunte a nuestro display. Por ejemplo:

```
$ xeyes -display joesws:0 -geometry 1000x1000+0+0
$ rsh big xterm -display myws:0 -ls </dev/null &
```

Control de acceso

El hecho de poder redirigir la salida de una aplicación X a través de la variable DISPLAY o de la opción `-display` no significa que siempre tengamos garantizado el acceso para realizar esto. De esta forma se evita que otros usuarios abran aplicaciones de forma indiscriminada en nuestro display. X implementa diversos mecanismos de control de acceso que pueden consultarse en la página de manual **X(1)**. El más simple y práctico emplea el comando **xhost**:

```
$ xhost +joews
$ xhost +
```

El primer ejemplo permite el acceso a las aplicaciones que se ejecuten en **joews** (independientemente del usuario que las ejecute), mientras que el segundo deshabilita el control de acceso por completo y cualquiera puede conectarse. Si se emplea **xhost** a secas se mostrará una lista de las máquinas que tienen permitido el acceso. Se puede retirar el acceso a alguna de

ellas empleando “**xhost -joe**ws”. Este comando suele emplearse cuando se desea ejecutar aplicaciones en un ordenador remoto al que se accede mediante *telnet*, por ejemplo. Una vez permitido el acceso podremos ejecutar aplicaciones X tras haber definido la variable DISPLAY.

5 Tipos de Sesión X. Inicio de sesión.

La forma de iniciar una sesión X depende principalmente de si accedemos al sistema a través de la propia consola del ordenador (estación de trabajo, etc.) o bien si lo hacemos a través de un terminal X.

5.1 Acceso a través de la consola

En este caso, nos encontramos ante la pantalla y el teclado del ordenador, que nos muestra una interface tipo texto. Lo primero es entrar en el sistema (hacer login) de la forma habitual. Para usar el entorno X tendremos que arrancar el servidor X y algún o algunos clientes, típicamente un emulador de terminal (**xterm(1)**) que nos permita seguir suministrando comandos al sistema una vez iniciada la sesión X. El programa diseñado para arrancar una sesión X es **xinit(1)**, cuya función es, básicamente, arrancar un servidor y un cliente, y esperar a que dicho cliente finalice para cerrar la sesión. **xinit** ofrece pocas facilidades para el usuario, por lo que habitualmente no es invocado directamente, sino a través de un *script* que se encarga de una inicialización útil y completa del entorno. Este *script* suele tener un nombre como **startx** y suele realizar funciones como:

- Inicializa el servidor más adecuado
- Carga el mapa del teclado
- Carga la configuración de los clientes (*Xresources*)
- Arranca los clientes preferidos por el usuario (u otros en su defecto)
- Arranca el window manager del usuario
- etc.

Scripts de este tipo son creados habitualmente por el super-usuario para controlar de que forma se inicia la sesión X y realizar las comprobaciones que se crean necesarias.

5.2 Acceso a través de un X-Terminal

La diferencia fundamental con el caso anterior es que no necesitamos arrancar el servidor X, ya que este ha sido arrancado al encender el terminal. Las tareas a realizar son:

- Dar acceso al usuario (login)
- Las funciones de **startx**, salvo la de inicializar el servidor.

Estas tareas suelen llevarse a cabo por un programa de gestión de la sesión como **xdm**, **gdm** o **kdm**, previamente configurado por el super-usuario, liberando al usuario de cualquier tarea manual.

6 Configuración de la Sesión X

En cualquiera de los dos casos (acceso por consola o por X terminal) el usuario dispone de una

serie de ficheros de configuración propios con los que puede hacer que el sistema se ajuste a sus preferencias. Estos ficheros suelen ser:

- **\$HOME/.xserverrc**: Servidor X que se desea arrancar (en acceso por consola). No suele ser necesario ya que el super-usuario ha debido establecer el servidor más adecuado por defecto.
- **\$HOME/.xinitrc**: Clientes que se deben ejecutar al arrancar la sesión X. Los clientes deben ejecutarse en *background* para impedir que el script se detenga en caso de que falle alguno de los clientes. El último cliente a ejecutar suele ser el window manager, que debe ejecutarse en *foreground*, ya que la finalización de este script determina el cierre de la sesión X.
- **\$HOME/.xsession**: En algunos sistemas se emplea este fichero como complemento al anterior. Habitualmente es éste el único que el usuario puede desear modificar. Un ejemplo de este fichero es el siguiente:

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

- **\$HOME/.Xresources**: Establece parámetros y opciones de diferentes clientes X.
- **\$HOME/.Xmodmap**: Mapa del teclado. Reasigna significados a las teclas o configura teclados con características especiales.

Si alguno de estos ficheros no está presente, el sistema leerá un fichero de configuración general en su lugar. Si los ficheros de configuración generales están bien contruidos, los usuarios no tienen por qué disponer de ficheros de configuración propios, a menos que quieran disponer de configuraciones ajustadas a su medida. En este caso, un buen punto de comienzo puede ser copiar el fichero de configuración general a su propio directorio raíz y hacer modificaciones sobre él. Algunos sistemas de escritorio modernos (KDE, GNOME) disponen de gestores de arranque que sustituyen a **xdm** y permiten cierta personalización de forma gráfica sin necesidad de modificar estos ficheros.

7 El Window Manager

Como mencionamos anteriormente, el entorno X de por si no impone una determinada decoración de las ventanas o un estilo para los menús. Es un programa ejecutado por el usuario, el *window manager*, el encargado de realizar estas operaciones. Existe gran diversidad de ellos disponibles. Algunos de los más clásicos son: Motif Window Manager (**mwm**), Open Look Window Manager (**olwm**) y Tab Window Manager (**twm**). Otros más modernos incluyen F? Virtual Window Manager (**fvwm**, **fvwm95**), Ice Window Manager (**icewm**), Window Maker (**wmaker**) y Enlightenment. Cualquier window manager ofrece las siguientes facilidades:

- Decoración de ventanas que puede incluir barras de título, marcos, botones, etc.
- Operaciones sobre las ventanas: mover, cambiar el tamaño, elevar, descender, iconificar/deiconificar, cerrar,...
- Menús desplegables: permiten el acceso rápido a las aplicaciones o utilidades preferidas por

el usuario.

- Máxima configurabilidad: colores, elementos decorativos, acciones del ratón, teclas de modo abreviado,...

El WM suele ser el último programa ejecutado durante la inicialización del entorno. De esta forma, salir del WM implica salir del entorno y cerrar la sesión X. Los distintos WM, al ser programas independientes, también tienen un modo de configuración particular en cada caso. Algunos incluyen una utilidad de configuración gráfica, como Window Maker o Sawfish, mientras que para configurar otros es necesario editar algún fichero de configuración (fvwm).

8 Clientes X y Recursos X.

Existe un gran número de Clientes X (programas que se ejecutan en entorno gráfico): editores de texto (**xedit**), lectores de correo electrónico (**xmail**), emuladores de terminal (**xterm**), visualizadores WEB (**netscape**), etc. Todos ellos admiten opciones en la línea de comandos al igual que los programas que se ejecutan en modo texto, pero el sistema X proporciona además un método estándar para que el usuario defina las opciones que debe emplear cada cliente X: los *Recursos X* o *X Resources*. Estas opciones consisten típicamente en el color de fondo o de primer plano, el tamaño de letra, el tamaño de la ventana, etc. Alternativamente, entornos de escritorio como GNOME o KDE proporcionan sus propios mecanismos de configuración del aspecto de las aplicaciones gráficas desarrolladas para estos entornos.

Los recursos X se especifican en ficheros de configuración tanto propios del usuario como del sistema en general. Estos ficheros, a su vez, pueden ser específicos de una aplicación o contener recursos para varias aplicaciones. En todos los casos, la forma de especificar recursos se ajusta al mismo formato, que se describe en **x(1)**. De esta forma, los lugares donde pueden encontrarse recursos son:

- Fichero de recursos del sistema (ej.: **/usr/lib/X11/sys.Xresources**): Es leído durante la iniciación de X mediante el programa **xrdb(1)** y contiene recursos para todo tipo de aplicaciones. Estos recursos son almacenados en el servidor, por lo que cualquier aplicación puede emplearlos, independientemente de si se trata de un cliente local o remoto.
- Fichero de recursos del usuario (ej. **\$HOME/.Xresources**): Tiene las mismas características que el anterior. Se lee durante la inicialización después del fichero de recursos del sistema, con lo que el usuario puede especificar sus propias opciones si lo desea.
- Ficheros de recursos del sistema específicos para cada aplicación: Consiste en un conjunto de ficheros, uno por aplicación, aunque no todas las aplicaciones han de tener uno. Es frecuente encontrarlos en el directorio **/usr/lib/X11/app-defaults/**. Suelen suministrarse con la distribución de cada aplicación y se leen al ejecutar el cliente en cuestión. De esta forma se libera al servidor de tener que cargar todos los recursos de todas las aplicaciones al iniciar la sesión. Ejemplos de ficheros de recursos específicos son **XTerm**, **XCalc** y **XClock**.
- Ficheros de recursos del usuario específicos para cada aplicación: Igual que los anteriores, pero propios de cada usuario. Si el usuario lo desea puede tener sus propias versiones de los ficheros de recursos específicos. En este caso debe situarlos en un subdirectorios llamado, típicamente, **\$HOME/.app-defaults**.

A menos que se quiera cambiar por completo el aspecto de una aplicación, el único fichero que el usuario tendrá que editar es **\$HOME/.Xresources**. Adicionalmente, el usuario puede

añadir, cambiar o eliminar recursos en cualquier momento con la utilidad **xrdb(1)**.

A continuación se muestra un ejemplo de fichero de recursos X.

Ejemplo de fichero **\$HOME/.Xresources**:

```
!Define el background por defecto para todos los clientes

*background:floralwhite

!Recursos para aplicaciones diversas

XPostit.saveNotes:True
XPostit*background:Yellow
XPostit.geometry:-110+0

Xman.manualBrowser.geometry:750x850

XClock*analog:true
XClock*update:60

Netscape.geometry:700x720
XClipboard.geometry:500x140

!***** Recursos para xterm *****

XTerm*BoldFont: 7x13bold
XTerm*Font: 7x13bold
XTerm*SaveLines:500
XTerm*JumpScroll:on
XTerm*ScrollBar:on
XTerm*ReverseWrap:on
```

9 Opciones en Línea de Comandos

La mayoría de los programas X intentan usar los mismos nombres para los argumentos y opciones en línea de comandos. Como ejemplo, todas las aplicaciones escritas con el *X Toolkit Intrinsics* aceptan automáticamente las siguientes opciones:

- **-display** *display*: Especifica el nombre del servidor X a emplear.
- **-geometry** *geometry*: Especifica el tamaño inicial y la localización de la ventana.
- **-bg** *color*, **-background** *color*: Especifica el color a emplear para el fondo de la ventana
- **-bd** *color*, **-bordercolor** *color*: Especifica el color a emplear para el borde de la ventana
- **-bw** *número*, **-borderwidth** *número*: Especifica la anchura en pixels del borde de la ventana.
- **-fg** *color*, **-foreground** *color*: Especifica el color del texto o de los gráficos.
- **-fn** *font*, **-font** *font*: Especifica el font (tipo de letra) para visualizar texto.
- **-iconic**: Indica que el usuario prefiere que la ventana de la aplicación no sea visible, como si el usuario la hubiera iconificado. Algunos WM pueden no hacer caso de esta opción.
- **-rv**, **-reverse**: Indica que el programa debe emular vídeo inverso si es posible, frecuen-

temente intercambiando los colores de fondo y de primer plano. No todos los programas la implementan.

- **+rv**: Indica que el programa no debe emular video inverso. Se usa para anular opciones por defecto, ya que el video inverso no siempre funciona correctamente.
- **-title string**: Especifica el título a usar para la ventana. Este título suele ser empleado por el WM para dar una indentificación a la ventana.
- **-xrm resourcestring**: Especifica una opción en formato de línea de recursos X. Tiene más prioridad que los recursos establecidos por defecto y es muy útil para especificar opciones que no están disponibles en línea de comandos.

10 Entornos de escritorio

La gran versatilidad de X permite la existencia de muy diversos programas gráficos, a menudo basados en librerías diferentes que le confieren a cada uno un aspecto y manejo particular. Un ejemplo es el gran número de *window managers* existentes, cada uno con un sistema de configuración y de manejo diferente. Esta gran diversidad puede llegar a ser incómoda para el usuario principiante que encuentra una gran diversidad de opciones y se siente incapaz de elegir, o bien, preferiría manejar un conjunto de aplicaciones con una interfaz similar para todas ellas.

Con el fin de proporcionar al usuario un entorno de trabajo homogéneo, se han venido desarrollando proyectos abiertos para crear lo que se ha dado en llamar entornos de escritorio (*desktop environments*). Los escritorios más populares en la actualidad son KDE y GNOME. Ambos se basan completamente en software libre (GPL). El proyecto GNOME ha sido adoptado por el proyecto GNU.

Estos entornos proporcionan un conjunto de aplicaciones básicas que comparten un aspecto similar. Así mismo incluyen un sistema de configuración de los diferentes componentes que resulta homogéneo y fácil de usar. La finalidad es que el usuario pueda realizar su trabajo cotidiano empleando las herramientas incluidas en el escritorio y se evite tener que aprender diferentes entornos y diferentes tareas de configuración.

11 Referencias

- X Consortium: www.x.org
- Proyecto XFree86: www.xfree86.org
- FreeBSD: www.freebsd.org
- GNU/Linux: www.linuxdoc.org
- Licencia GPL: www.gnu.org/home.es.html
- KDE: www.kde.org
- GNOME: www.gnome.org