Las interfaces gráficas en Linux

ovtoaster.com/interfaces-graficas-en-linux/

AsierPH 1/17/2015

En el mundo Linux existe una gran variedad de entornos gráficos, siendo para algunos una bendición de posibilidades y para otros un dolor de cabeza de fragmentación. En este artículo no voy a entrar en valoraciones sobre si es bueno o malo que existan tantas alternativas, simplemente voy a limitarme a hablar de las diferentes capas que componen las interfaces gráficas en Linux y su funcionamiento, de forma que se entienda porque es posible que existan tantas opciones.

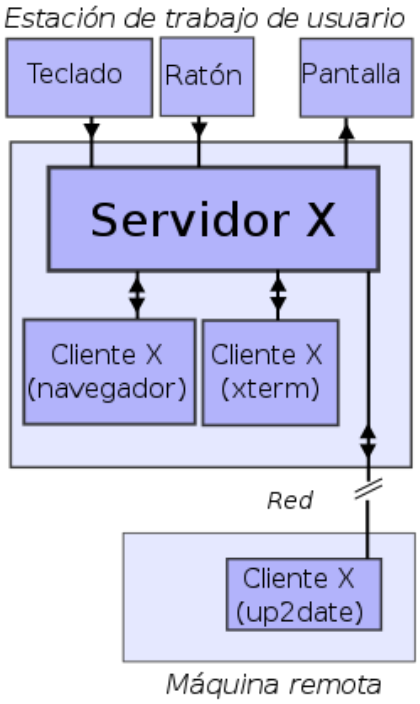
X Window system, la base.

Mientras que en otros sistemas operativos (Windows de Microsoft por ejemplo) la interfaz gráfica viene integrada en lo más profundo del sistema, en Linux y Unix se utiliza un **software independiente** al sistema operativo. Este software que fue desarrollado por el MIT en la década de los 80, se denomina **X Window System**, o de forma más corriente, X Window o X11 (debido a que se encuentra en la versión 11).

X Window es un sistema gráfico completo que se encarga de dibujar y gestionar los eventos de los componentes de un entorno gráfico: ventanas, botones, menús, listas, el cursos, etc. Este sistema de gráficos es la capa más baja de la interfaz gráfica en Linux y gestiona las interacciones entre máquina y humano, además de dibujar los componentes gráficos que se le indica. Pero **X Window no contiene los componentes** (el aspecto) a dibujar, estos son ofrecidos por el gestor de ventanas del que hablo más adelante.

El modelo cliente/servidor

X Window funciona bajo el modelo cliente/servidor, ¿ Pero cómo es esto?



En X window existe un **servidor** que se encarga de proporcionar herramientas para acceder a la pantalla, teclado y ratón, es decir, un ordenador corriente que disponga de estos periféricos. Y por otra parte están los **clientes**, aplicaciones que utilizan estos recursos para interaccionar con el usuario. Normalmente tanto el cliente como el servidor están en un mismo ordenador, pero X Window ofrece la posibilidad de realizar la comunicación vía red gracias a un protocolo denominado Xprotocol y la librería Xlib. Las aplicaciones son clientes porque el servidor X les provee “servicio de ventanas”.

El proceso se efectúa de la siguiente manera: Cuando un sistema Linux con interfaz gráfica inicia, lo hace con el servidor X, cada vez que se inicia una aplicación se crea un cliente que se comunicara con el servidor para ofrecer una interfaz gráfica por pantalla. Por otra parte al interactuar con el ratón o el teclado, el servidor enviara la señal al cliente (aplicación) y ésta ejecutara las acciones pertinentes, después se enviara el resultado al servidor para que sea visualizado en la pantalla.

Ejemplo: click con el mouse en un menú (servidor) envía señal al programa

(comunicación servidor-cliente) el cliente realiza la operación de abrir el menú (cliente) le envía la señas de visualización al servidor (comunicación cliente-servidor) el servidor lo muestra en la pantalla (servidor)

Los gestores de ventanas

X11 solo se encarga de dibujar la ventana, pero no sabe como renderizar los diferentes elementos que la componen. Para este cometido existe **el gestor de ventanas** o **window manager** que se encarga de indicar al servidor X como dibujar los diferentes elementos que requiere una ventana: movimientos, selecciones, elementos y decoraciones. En resumidas cuentas, **el Servidor X visualiza lo que este gestor le indica**.

En el mundo del pingüino existe una gran variedad de gestores de ventanas, la diferencia entre estos radica en diferentes aspectos como pueden ser la apariencia visual, el consumo de memoria, las opciones de personalización, etc.

Lista de gestores de ventanas populares <Wikipedia>

- AfterStep, basado en FVWM y de apariencia similar a NeXTSTEP. Página Oficial de AfterStep
- AmiWM (Amiga Window Manager). Página Web de AmiWM
- Blackbox. Sitio Oficial de Blackbox
- CTWM. Sitio Web de CTWM
- Enlightenment (también llamado 'E'), basado originalmente en fvwm2. Sitio Oficial de Enlightement
- Fluxbox, derivado de la versión 0.61.1 de Blackbox. Sitio Web de Fluxbox
- FVWM. Página Oficial de FVWM
- FVWM95, versión modificada de fvwm2.x para que tome el aspecto de Windows 95.
- IceWM. Sitio Oficial de IceWM
- Ion WM
- Kwin, gestor de ventanas de KDE.
- Metacity, el gestor de ventanas ligero de algunas versiones de GNOME 2.
- Metisse, gestor de ventanas en 3D basado en otro gestor de ventanas, FVWM),
- Motif (Motif Window Manager).
- OLWM/OLVWM (OpenLook Window Manager / OpenLook Virtual Window Manager). Página Oficial de Olvwm
- Openbox, inicialmente basado en Blackbox y luego reescrito de cero, con varias ventajas incluyendo fuentes 'anti-alising'.
- quartz-wm, gestor de ventanas de Apple, de aspecto similar a Aqua, para el sistema X Window (X11) en Mac OS X.
- Sawfish, originalmente conocido como Sawmill. Sitio Oficial de Sawfish
- SCWM. Sitio Oficial de SCWM
- TWM/ VTWM (Tab Window Manager, también llamado Tom's Window Manager / Virtual TWM).
- WindowMaker, emula la interfáz de NeXT, como AfterStep. Sitio Oficial de WindowMaker
- wm2/wmx. Página de wm2 y de wmx

Widgets y toolkits

Los gestores de ventanas son algo primitivos por si solos, y les falta una mayor integración de los diferentes programas en el entorno. Para poder integrar de manera adecuada los programas, existen diferentes librerías que se ocupan de esto. Teniendo en cuenta que a los elementos que componen una interfaz gráfica (menús, botones, campos de entradas, etc) se les conoce como **widgets**, las librerías se encargan de proporcionar funciones para gestionar y dibujar estos widgets. Cuando una de estas librerías contiene un kit completo de widgets, se le denomina **widget toolkit** o simplemente **toolkit**. Existen diferentes toolkits, pero los dos más utilizados son:

GTK (Gimp toolkit) : En un principio se escribió para el programa GIMP de retoque de imágenes, pero con el tiempo se a vuelto independiente a este gracias a su eficacia. Está escrito en C y es la librería por defecto del conocido entorno gráfico GNOME.

QT (Cute) : Esta escrita en c++ pero tiene compatibilidad con muchos otros lenguajes. QT además de ser una toolkit, ofrece un entorno de desarrollo de aplicaciones. KDE utiliza QT como librería por defecto.

Estas dos librerías no son compatibles de manera nativa, pero es posible abrir programas desarrollados en una en sistemas que usen por defecto la otra. Simplemente la visualización cambiará.

Los entornos de escritorio o gráficos.

Por ultimo tenemos los entornos de escritorio, **se basan en los toolkits** y ofrecen una solución completa de interfaz gráfica. Un entorno de escritorio provee al usuario de iconos, barras de herramientas, aplicaciones e integración entre aplicaciones con habilidades como arrastrar y soltar que hacen un manejo más amigable. Unos de los más usados son :

Existen muchos más, y los últimos años han aparecido nuevos entornos de todo tipo, pero hablar de todos ellos llevaría un artículo a parte.

El futuro, Wayland

Una vez comentadas todas las capas que componen las interfaces gráficas en Linux y hablado de como funcionan e interactúan entre ellas, es conveniente hablar de lo que depara el futuro, y parece que el futuro depara Wayland.

Wayland es un sistema de gráficos **basado en OpenGL** que empezó a desarrollarse en 2008 por el ingeniero de software Kristian Høgsberg para sustituir a X Window System . Pero la pregunta clave es ¿Hace falta remplazar X Window System?

X window lleva desde hace años acarreando una serie de problemas que se han solucionado con parches sobre parches, **convirtiendo X11 en un sistema muy pesado** y por lo tanto, menos eficiente. Los problemas técnicos de X Window se analizan en este artículo (en ingles) de forma muy clara y concisa.

Y es en el contexto de estos problemas donde entra Wayland, que **esta escrita con un diseño más eficiente y limpio, adaptándose a los sistemas gráficos modernos**. Las principales ventajas de Wayland según sus desarrolladores son:

Cada frame es perfecto, sin retardos, lag o cualquier efecto negativo en la visualización, sin importar la carga del sistema.

Sistema mínimo, sin cosas innecesarias que solo afectan negativamente en el rendimiento y no sirven para nada.

Posibilidad de crear **versiones especificas** para un tipo de hardware, pero sin ser necesario.

En este artículo se exponen todas las ventajas que trae consigo Wayland. Por otra parte, entre los desarrolladores de Wayland están algunos de los antiguos desarrolladores de X Window, que conocen perfectamente los problemas de X11 y por lo tanto, saben a que se enfrentan.

Fedora, KDE y otros proyectos Open Source ya han manifestado sus intenciones de migrar a este nuevo sistema y puede que en en los siguientes años otros proyectos tomen el mismo rumbo.

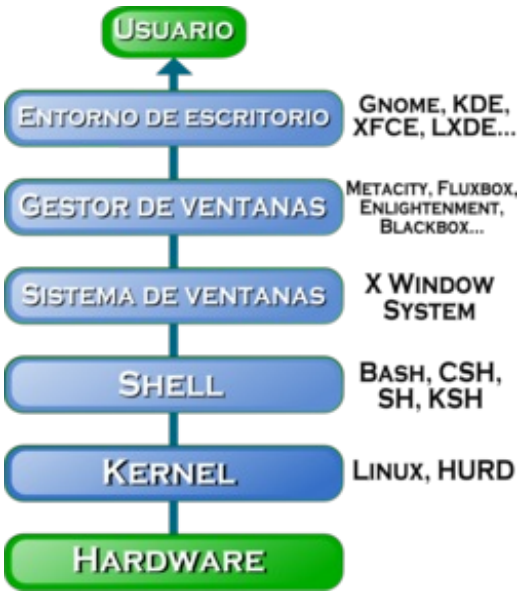
Dicho esto, también existen detractores que creen que todo queda muy bonito sobre el papel pero que a la hora de implementarlo el rendimiento no sera el mismo del que se habla o que sera muy similar al de X11. Además tanto Nvidia como ATI han dicho que de momento no esta entre sus planes adaptar sus drivers a Wayland, siendo este un gran inconveniente. Por ultimo, es importante decir que **X Window es un sistema completamente testeado y estable**, mientras que Wayland, no.

Personalmente creo que cuantas mas opciones mejor, y si encima son opciones innovadoras que modernizan sistemas arcaicos, mejor que mejor. Aun así abra que ver como evoluciona el proyecto y si al final realmente cumple con los que se dice, y más importante, si los proyectos open source deciden adoptarlo.

Conclusión

Como hemos visto, las interfaces gráficas en Linux están compuestas por diferentes capas, en la siguiente imagen se pueden observar estas capas y la interacción entre ellas:

«Esquema de las capas de la interfaz gráfica de usuario» por Noelia Sales Montes



Lo fantástico de abstraer en diferentes capas la GUI, es que es posible crear y utilizar diferentes interfaces y permite mucha mayor adaptación a cada persona o situación. Pero también tiene parte negativa, el desarrollo de aplicaciones se hace específicamente para un tipo de toolkit, dando incompatibilidades y en el caso de los usuarios nuevos tener diferentes tipo de interfaces puede ser un poco complicado.

¡Espero que os sea de ayuda!