

Resumen Ponencia : Nueva generación de gráficos en Linux

Franco Catrin L.

El objetivo de la ponencia es mostrar los distintos avances que se están realizando para hacer un sistema gráfico más flexible y avanzado para Linux, aprovechando al máximo las actuales capacidades del hardware. Estos cambios cubren varias capas del sistema gráfico, desde bibliotecas para ser usadas por los programadores, hasta nuevas formas de interactuar con el hardware.

A medida que ha ido mejorando el hardware, se han ido mejorando las capacidades gráficas de Linux, probablemente los usuarios nuevos no lo conocen, pero el escritorio Linux de hoy es muy distinto al que teníamos solo hace unos 5 años atrás. Aceleración 3D, fonts suavizados, fonts bidireccionales, son características que no existían en esos años, o recién se estaban desarrollando. Hoy en día también hay nuevas características que están en pleno desarrollo, pero que en un tiempo más serán algo común.

A continuación se describen los principales componentes que se describirán en mi presentación, se incluirán algunas demos para que los asistentes disfruten "en vivo" estas nuevas características.

Xrender

Xrender es una Extension para X que surge debido a una necesidad que había de poder dibujar texto u otro tipo de figuras con anti-alias (suavizado). La extensión permite que el servidor X sea capaz de realizar operaciones gráficas complejas, como mezclas de texturas o alpha blending. Aunque aun no es una práctica común. La aceleración por hardware de Xrender permite que el servidor X pueda realizar eficientemente las operaciones definidas por la extensión sin malgastar ciclos de CPU, y aprovechando las facilidades que ofrecen los procesadores gráficos actuales.

Cairo

La biblioteca Cairo permite que las aplicaciones puedan realizar operaciones gráficas complejas, de una forma simple e independiente de los dispositivos. Por ejemplo, cuando una aplicación dibuja un polígono 2D, la biblioteca es capaz de realizar operaciones de suavizado de los bordes de forma transparente para la aplicación.

La biblioteca esta pensada con la extensión Xrender en mente, e implementa el modelo PDF 1.4 que incluye interesantes y complejas operaciones gráficas.

Una característica importante de cairo, es que se presenta como una API única para las aplicaciones, pero el destino del render puede ser cambiado, de esta forma, por medio de una misma API se puede dibujar sobre imágenes png, postscript, pdf, svg, xlib y glitz. Este último destaca por implementar Cairo usando OpenGL, de esta forma se utilizan el poder de los nuevos procesadores gráficos 3D para renderizar imágenes 2D.

La próxima versión de GTK+ (2.8) utilizará Cairo como API de bajo nivel, esto significará un cambio radical en cuanto a la capacidad del toolkit para dibujar las aplicaciones, se espera que aparezcan nuevos themes que puedan aprovechar al máximo estas nuevas capacidades gráficos.

(Demo de Cairo)

Extension Composite

La forma en que se dibujan las ventanas en el sistema actual es en base al algoritmo del pintor. Este algoritmo simplemente dibuja una ventana sobre la otra de tal forma que las ultimas ventanas dibujadas cubren zonas de las ventanas que ya fueron dibujadas. Esto era ineficiente para las características del hardware que existía antes, por lo tanto se utiliza clipping.

Con clipping, cuando una ventana se mueve, y deja visible un área de otra ventana, esta área descubierta es redibujada por la aplicación, es fácil notar esta operación al mover una ventana sobre otras ventanas complejas, se puede ver como cada aplicación trata de dibujar sus áreas descubiertas. Como cada área se dibuja solamente una vez, no se pueden implementar características como transparencias o sombras, ya que en realidad bajo una ventana no se dibuja nada.

La extensión Composite cambia el modelo, considerando que las nuevas tarjetas de video tienen mejores características. Lo que hace la extensión es que cada ventana se dibuja completamente en memoria, es decir no hay áreas ocultas de ventanas, y estas se dibujan solamente una vez (a menos que se modifiquen). Una aplicación especial del sistema, llamado Composite Manager es la encargada de tomar estas ventanas y componerlas en el área visible de la memoria de video. Al realizar esta operación, se pueden realizar operaciones de blending entre las distintas ventanas, por ejemplo para dibujar sombras proyectadas entre una ventana y otra, facilitando la visualización de profundidad por parte del usuario.

Otro posible uso de esta extensión es la mejora en las características de accesibilidad del escritorio por personas con dificultad visual, por ejemplo en vez de dibujar la ventana del mismo tamaño de la original, se puede ampliar con ayuda del procesador gráfico.

También la extensión permite disponer libremente las ventanas en pantalla, por ejemplo una ventana se puede mostrar en un ángulo distinto, o incluso disponer en forma 3D como lo hace el proyecto Looking Glass de Sun.

Como las ventanas que están dibujadas off-screen, estas pueden ser copiadas las veces que se desee en el área visible de la pantalla. Por ejemplo, con Composite se puede hacer un visualizador de ventanas iconificadas, o ventanas de otro escritorio virtual que contenga el contenido real de las ventanas, incluso si se están dibujando frecuentemente como es el caso de un reproductor de video.

Las potencialidad de Composite aun se esta descubriendo.

X Composite Manager

Se trata de una aplicación de pruebas de la extensión Composite, esta aplicación implementa transparencias, sombras y fading de ventanas, cambiando radicalmente la forma en que se despliega el escritorio de Linux.

(Demo de X Composite Manager)

Luminocity

Luminocity es una aplicación de prueba para integrar características de Composite Manager en el gestor de ventanas Metacity. Luminocity se apoya en GLX para hacer el render de las ventanas, y se pueden obtener resultados bastante interesantes

(Demo de Luminocity)

Extended Acceleration Architecture (XAA)

XAA o eXtended Acceleration Architecture fue introducido en la versión 4 del Xfree86. XAA es una API que deben implementar los drivers de video para acelerar operaciones 2D. Si bien XAA fue una solución para los años en que fue liberada esa versión de Xfree86, no es una buena solución para los requerimientos actuales de los gráficos en Linux, específicamente la extensión Xrender. Incluso sin utilizar aceleración por hardware, pero utilizando otra arquitectura de aceleración (como kdrive), es posible obtener mejores resultados que en un hardware acelerado utilizando XAA.

EAX

EAX es una nueva arquitectura de aceleración, que toma como base Kdrive, pero busca acelerar completamente Xrender. Actualmente se esta implementando EAX en todos los drivers que soportan XAA, para que soporten ambas arquitecturas de aceleración, la idea es que XAA deje de ser utilizado para que solamente quede EAX, esto significa Xrender acelerado, por lo tanto las aplicaciones que usen Cairo, y Composite Managers como X Composite Manager tambien funcionarán con ayuda del hardware.

Sin embargo, hay algunos que no creen que EAX sea una solución en el largo plazo, sino que se debe utilizar algo basado en OpenGL, considerando que EAX es un subconjunto de las operaciones que permite OpenGL.

XGL

XGL es otra forma de interactuar entre las aplicaciones y el hardware. La idea es que X se implementa sobre un driver OpenGL, las operaciones que requieren las aplicación son renderizadas por OpenGL, que puede estar acelerado o bien puede tener funciones implementadas por Mesa cuando el hardware no lo puede realizar.

Al utilizar XGL, solo se podría utilizar con hardware actual, la excusa es que una tarjeta con buen soporte de OpenGL es bastante accequible hoy en día.

(Demo de XGL)

Referencias

<http://dri.freedesktop.org/~jonsmirl/graphics.html>

<http://live.gnome.org/Luminocity>

<http://www.gnome.org/~seth/blog/xshots>

http://www.osnews.com/story.php?news_id=9609