



Quartz Extreme

Acerca de este documento

En este reportaje vamos a ver que es Quartz Extreme, y que ventajas aporta.

Quartz Extreme, no es más que el nombre que Apple da a su nuevo gestor de ventanas acelerado por hardware que incorpora desde Mac OS X 10.2

Nota legal

Este tutorial ha sido escrito por Fernando López Hernández para MacProgramadores, y de acuerdo a los derechos que le concede la legislación española e internacional el autor prohíbe la publicación de este documento en cualquier otro servidor web, así como su venta, o difusión en cualquier otro medio sin autorización previa.

Sin embargo el autor anima a todos los servidores web a colocar enlaces a este documento. El autor también anima a cualquier persona interesada en conocer Quartz Extreme a bajarse o imprimirse este reportaje.

Madrid, Julio 2004

Para cualquier aclaración contacte con:

fernando@DELITmacprogramadores.org

Tabla de contenido

Tabla de contenido	3
Conceptos previos.....	4
Arquitectura del sistema de ventanas de Mac OS X	6
Quartz Extreme	8

Conceptos previos

Antes de explicar en que consiste Quartz Extreme, vamos a explicar primero como está implementado el sistema de ventanas de OS X.

El sistema de ventanas de Apple consta principalmente de cinco subsistemas gráficos:

Quartz 2D. Cuyo nombre formal es "Core Graphics Rendering" es una librería de dibujo en 2D que acepta instrucciones de dibujo en forma de figuras lógicas (líneas, círculos, botones, comboboxes, texto...) y transforma estas entradas en una salida en varios formatos de salida (PostScript, PDF y por supuesto bitmaps). En el resto de este artículo supondremos que la salida la genera en forma de bitmap, pero el lector debe recordar que la salida también la puede generar en forma de, por ejemplo, PDFs, lo cual es muy útil para imprimir esas salidas.

En el documento oficial de Apple "Mac OS X System Overview", este subsistema se describe así:

[Quartz 2D's] APIs allow you to create text and images by specifying a sequence of commands and mathematical statements that place lines, shapes, color, shading, translucency, and other graphical attributes in two-dimensional space. You do not need to specify the attributes of individual pixels. As a result, a shape can be efficiently defined as a series of paths and attributes rather than as a bitmap.

QuickDraw. Es la antigua librería gráfica de Mac OS Classic, aunque en OS X esta librería en gran parte se limita a llamar a Quartz 2D para que le realice las operaciones de dibujo.

OpenGL. Es la librería de dibujo 3D de OS X. Esta librería se basa en el estándar OpenGL de Silicon Graphics International (SGI). Otros muchos sistemas operativos la utilizan, una excepción destacable es Windows que tiene su propia librería de dibujo llamada Direct X (aunque lógicamente podemos instalar OpenGL en Windows como una aplicación más).

QuickTime. La librería de Apple para producir video de alta calidad.

Quartz Compositor. Todos los datos en forma de bitmaps que producen Quartz 2D, QuickDraw, OpenGL y QuickTime se pasan a Quartz Compositor para ser mostrados en la pantalla. Es decir, Quartz Compositor (formalmente llamado "Core Graphics Services") está implementado como un servidor de ventanas implementado en un sólo proceso y es el responsable de manejar todas las ventanas de la pantalla.

Cada ventana tiene asociado un bitmap (creado por la aplicación dueña de la ventana) y que modifica usando las APIs de dibujo. El servidor de ventanas

produce el resultado final en pantalla componiendo todos los bitmaps de las diferentes ventanas de acuerdo a las coordenadas y capa que ocupan.

En concreto en "Mac OS X System Overview" se describe como:

[The window server] composites and recomposites each pixel of an application's window as the window is drawn, redrawn, covered, and uncovered. Each window is represented as a bitmap that includes both translucency (alpha channel) and anti-aliasing information. The bitmap serves as a buffer, allowing the window server to "remember" an application's window contents and to recomposite it without the application's involvement.

Otra misión del gestor de ventanas es enviar los eventos (movimientos y pulsaciones de ratón así como pulsaciones de teclado) a su correspondiente aplicación, así como gestionar el cursor del ratón.

Es decir, en resumen, la aplicación envía comandos de dibujo a las diferentes APIs, estas APIs producen bitmaps, y el servidor de ventanas (Quartz Compositor) combina estos bitmaps para obtener el contenido final de la pantalla.

Si este documento le está resultando útil puede plantearse el ayudarnos a mejorarlo:

- Anotando los errores editoriales y problemas que encuentre y enviándolos al sistema de Bug Report de Mac Programadores.
- Realizando una donación a través de la web de Mac Programadores.

Arquitectura del sistema de ventanas de Mac OS X

Existe un problema importante de rendimiento con la arquitectura de Mac OS X, que es que la cantidad de memoria que usa el servidor de ventanas crece linealmente con el número de ventanas que tenemos abiertas.

En arquitecturas más tradicionales (como puedan ser el sistema de ventanas de Microsoft Windows o el de XWindows) este problema no existe porque cuando estos sistemas se diseñaron el mantener en memoria un bitmap con la imagen de cada una de las ventanas era simplemente inviable. En estas arquitecturas, lo que se hace es que cada vez que el servidor de ventanas quiere dibujar una ventana, este manda una orden a la aplicación dueña de la ventana para que la redibuje. Obsérvese que aquí existe un trade-off entre velocidad y memoria, ya que esta solución tradicional aunque consume menos memoria, es más lenta.

En las arquitecturas tradicionales existe un frame buffer del tamaño y resolución de la pantalla (p.e. 1024x768 a 32 bits de profundidad de color) en el que todas las ventanas dibujan. Este frame buffer se almacena en la memoria de video.

Para evitar el molesto efecto que puede experimentar el usuario de ver como se dibuja en pantalla, se utiliza una técnica llamada **doblo buffer**, que consiste en dibujar primero en memoria y luego volcar este contenido a la memoria de video. Este doble buffer puede estar o bien en memoria convencional o bien en la memoria de video (si tiene suficiente memoria).

Según esto es fácil calcular la memoria que requiere un sistema de ventanas tradicional. P.e. 2 frame buffers x 1024 píxeles x 768 píxeles x 32 bits por píxel = 6.291.456 B, es decir, 6 MB.

Sin embargo, en la arquitectura de Mac OS X, el tamaño de la pantalla no determina la memoria necesaria, sino que lo determina el número de ventanas que tengamos creadas (que suele ser mucho mayor), cada una de ellas con su bitmap, que puede ser del tamaño de la ventana (o mayor si la ventana tiene scroll).

Hasta Mac OS X 10.1 estos bufferes de bitmap se almacenaban en memoria convencional, dejando sólo en memoria de video la memoria del servidor de ventanas. Con la llegada de Mac OS X 10.2, va a ser posible (si la tarjeta de video tiene suficiente memoria) mover los bufferes de ventana a la tarjeta de video, evitando todo el trasiego de información de memoria convencional a la tarjeta.

De aquí viene el hecho de que Quartz Extreme exija disponer de 16 MB de RAM en video (32 MB recomendables).

¿Qué ventana tiene entonces esta nueva arquitectura? Las ventajas son básicamente dos:

Por un lado se consigue mayor velocidad, siempre que haya memoria suficiente (esta ventaja se conseguía también en Mac OS X 10.1, ya que, aunque el bitmap estaba en memoria convencional, no había que repintar la ventana cada vez que se invalidaba).

Por otro lado se consiguen efectos gráficos que hasta ahora eran prohibitivos. Estamos hablando de las transparencias y sombreados (véase Figura 1), un efecto tan conocido como característico de OS X. Para ello, en vez de que el servidor de ventanas coja sin más los píxeles de las ventanas y los vuelque en la pantalla, el servidor de ventanas puede combinar los píxeles de varias ventanas teniendo en cuenta el factor de transparencia de cada ventana.



Figura1: Ejemplo de transparencias y sombreados en Mac OS X 10.2

Quartz Extreme

Hasta Mac OS X 10.1 Quartz Compositor operaba completamente por software usando la CPU (Central Process Unit) para posicionar las ventanas en capas y enviar el resultado final a la tarjeta de video, con lo que la tarjeta de video actuaba como un almacén temporal de la imagen que estábamos mostrando en pantalla.

Con la llegada de Mac OS X 10.2 y de Quartz Extreme, el gestor de ventanas se mueve al GPU (Graphics Process Unit), dejando a la CPU más libre para acometer otras tareas.

¿Qué es entonces Quartz Extreme?. Quartz Extreme no es más que una nueva implementación de Quartz Compositor usando OpenGL, en concreto usando una textura de OpenGL. Por eso a Quartz Extreme se le llama informalmente QuartzGL.

Cuando una aplicación usa una API de dibujo para emitir un comando de dibujo, la API produce un bitmap (posiblemente basado en una imagen vectorial), y la textura OpenGL, que no es más que una aplicación, coloca ese bitmap en la textura usando los comandos de dibujo de OpenGL.

De esta forma, el servidor de ventanas no es más que una aplicación OpenGL. Eso si, es la única aplicación OpenGL que no envía su salida al servidor de ventanas, porque esa aplicación es en si el servidor de ventanas.

La Figura 2 y Figura 3 muestran la diferencia entre Quartz Compositor antes y después de la llegada de Quartz Extreme. Obsérvese la proliferación de línea rojas (hardware) en Quartz Extreme. Con Quartz Extreme, lo único que no se ha pasado al GPU ha sido la implementación de Quartz 2D, QuickDraw y QuickTime, que sigue realizándose en la CPU (una posible mejora para futuras versiones). OpenGL si que utiliza la GPU de la tarjeta de video.

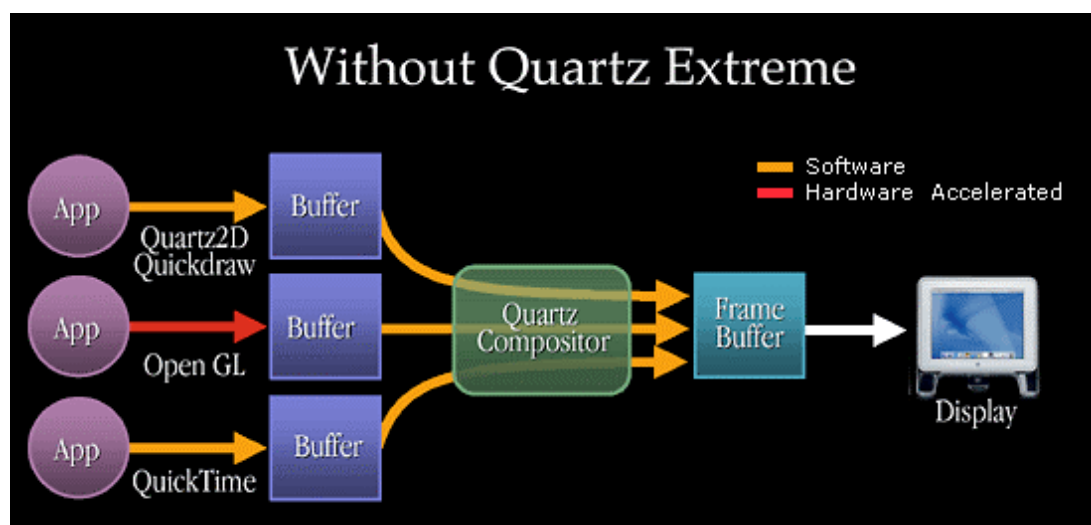


Figura 2: Sistema de video antes de Quartz Extreme

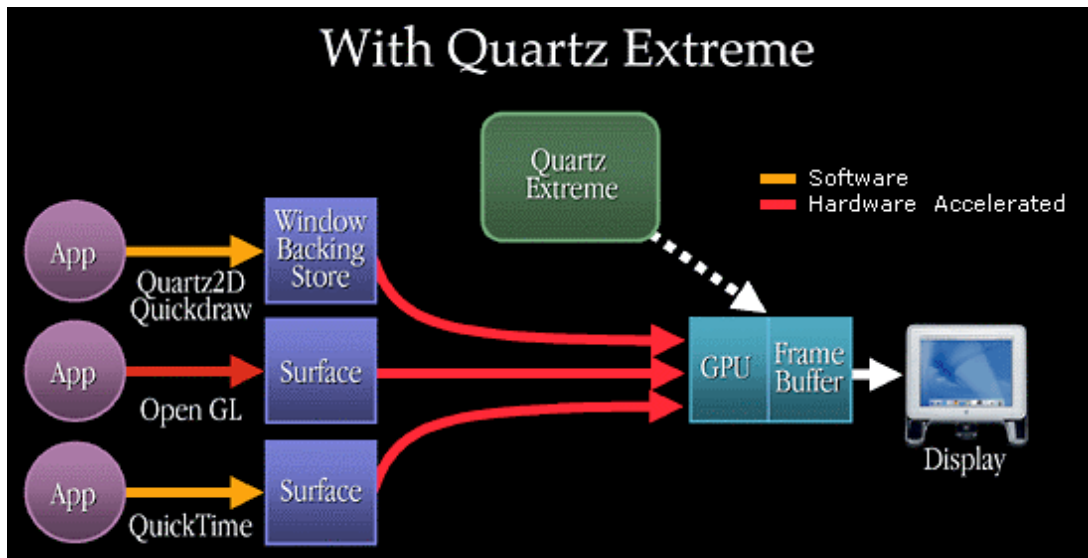


Figura 3: Sistema de video con la llegada de Quartz Extreme

Otra importante mejora que se aprecia en Mac OS X 10.2 ha sido respecto al scroll. Antes cuando hacíamos un scroll se repintaba el contenido de toda la ventana, es decir, se volcaba del buffer de la ventana a la memoria de video (era especialmente apreciable al ver páginas web con un browser). Jaguar simplemente deslaza el contenido de la memoria de la tarjeta de video (lo cual se podría haber hecho por hardware sin necesidad de Quartz Extreme), y después se mueve, de buffer de ventana a la memoria de video, sólo la nueva porción de la ventana.

Madrid, Julio 2004

Fernando López Hernández

fernando@DELITmacprogramadores.org