

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

SISTEMAS OPERATIVOS

***EL CAMINO DE LA GPU A TRAVÉS
DE LOS SISTEMAS OPERATIVOS***



MAESTRO: ING. GUNNAR EYAL WOLF ISZAEVICH

GRUPO: 06

INTEGRANTES:

GONZÁLEZ LÓPEZ DAVID

REYES ROMERO LUIS FERNANDO

SEMESTRE: 2024-2

FECHA DE ENTREGA: 12 DE MARZO DE 2024

Índice

- **Introducción**

- ¿Qué es la GPU?

- **Historia de la GPU**

- ¿Qué hubo antes de la GPU?

- La llegada de la GPU

- ¿Cómo se comunicaban la GPU y el SO?

- Framebuffers

- **La GPU y el SO**

- ¿Cómo se comunican?

- Elementos de comunicación

- **Referencias**

Introducción

¿Qué es la GPU?

Conocida como la Unidad de Procesamiento Gráfico (GPU por sus siglas en inglés) es un circuito o chip electrónico capaz de realizar cálculos matemáticos a gran velocidad. El diseño de una GPU le permite realizar la misma operación en múltiples valores de datos de forma paralela, esto mismo aumenta su eficiencia de procesamiento para muchas tareas de cálculo intensivo.

Al igual que una unidad central de procesamiento (CPU), una GPU también es un componente del chip de los dispositivos informáticos. Sin embargo, una diferencia importante es que la GPU está diseñada específicamente para manejar y acelerar cargas de trabajo gráficas y mostrar contenidos gráficos en un dispositivo como un PC o un smartphone, aunque con el tiempo, las mejoras tecnológicas han dado lugar a GPU más flexibles y programables que pueden utilizarse para muchas más aplicaciones.

Historia de la GPU

¿Qué hubo antes de la GPU?

Antes de la llegada de la GPU, el manejo y despliegue gráfico se hacía por medio de pantallas de impresión matriciales de puntos, que salieron al mercado en las décadas de 1940 y 1950 a la par de la computadora ENIAC, la cual fue la primer computadora de propósito general, cuyas salidas eran justamente mediante dispositivos de impresión con caracteres alfanuméricos los cuales el usuario examinaba. Tras esto, aparecieron las pantallas vectoriales y rasterizadas y, más tarde, las primeras consolas de videojuegos y los PC.

En aquella época, había un dispositivo no programable llamado controlador gráfico, el cual coordinaba la visualización gráfica per se en la pantalla. Los controladores gráficos dependían tradicionalmente de la CPU para el procesamiento, aunque algunos incluían procesadores en chip, que a pesar de ello, se encontraban sumamente limitados dada su baja capacidad de procesamiento y su ineficiencia al gestionar su despliegue gráfico.

Un ejemplo de lo limitantes que eran los primeros sistemas operativos al implementar una gestión de procesamiento gráfico, fue el CP/M originalmente

Control Program/Monitor y más tarde Control Program for Microcomputers, el cual tuvo como propósito el organizar archivos en un medio de almacenamiento magnético, y cargar y ejecutar programas almacenados en un disco. En los que, debido a la pequeña cantidad de memoria disponible, los gráficos nunca fueron una característica común asociada a los sistemas operativos CP/M de 8 bits. La mayoría de los sistemas sólo podían mostrar gráficos y diagramas rudimentarios de arte ASCII en modo texto o utilizando un juego de caracteres personalizado en un monitor CRT.

Los primeros verdaderos gráficos 3D se originaron con los primeros controladores de pantalla, conocidos como cambiadores de vídeo y generadores de direcciones de vídeo. Estos dispositivos actuaban como intermediarios entre el procesador principal y la pantalla. Convertían el flujo de datos entrante en una salida de vídeo en serie con mapa de bits, que incluía luminancia, color y sincronización vertical y horizontal compuesta. Esta sincronización era crucial para mantener la alineación de los píxeles en una generación de pantalla, garantizar la progresión ordenada de cada línea sucesiva y gestionar el intervalo de supresión (el tiempo entre el final de una línea de exploración y el comienzo de la siguiente).

A principios de los años 70, el desarrollo de chips de circuitos integrados de memoria MOS (memoria semiconductora de óxido metálico), en particular chips DRAM (memoria dinámica de acceso aleatorio) de alta densidad con al menos 1 kb de memoria, hizo posible crear, por primera vez, un sistema de memoria digital con framebuffers capaces de contener una imagen de vídeo estándar.

Los propios frame buffers son un elemento crucial en el despliegue de gráficos en pantalla, ya que estos sustituyen de forma digital el funcionamiento de las impresiones matriciales previamente mencionadas, esto debido a que son una parte de la computadora que guarda temporalmente una imagen para después ser mostrada en el monitor.

El frame buffer se utiliza principalmente para almacenar de forma temporal una imagen en la memoria antes de enviarla al monitor, ya que el procesamiento de una imagen en tiempo real puede ser demasiado intensivo para la CPU, por lo que es necesario almacenar la imagen en una ubicación temporal antes de enviarla al monitor.

De igual forma, existen algunos casos donde el adaptador de vídeo está integrado en la placa madre, provocando que el frame buffer sea almacenado en nuestra memoria principal (RAM).

La llegada de la GPU

Las modernas GPU son descendientes de los chips gráficos monolíticos de finales de la década de 1970 y 1980, como ya se ha visto. Estos chips tenían soporte BitBLT limitado, la cual es una operación de datos utilizada habitualmente en infografía en la que varios mapas de bits se combinan en uno solo mediante una función booleana, esta misma se presentaba en la forma de sprites, y usualmente no tenían soporte para dibujo de figuras.

Algunas GPU podían ejecutar varias operaciones en una lista de display y podían usar DMA para reducir la carga en el procesador anfitrión; un ejemplo temprano es el coprocesador ANTIC usado en el Atari 800 y el Atari 5200. Hacia finales de la década de 1980 y principios de la de 1990, los microprocesadores de propósito general de alta velocidad fueron muy populares para implementar las GPU más avanzadas. Muchas tarjetas gráficas para PC y estaciones de trabajo usaban procesadores digitales de señales (DSP por sus siglas en inglés) tales como la serie TMS340 de Texas Instruments, para implementar funciones de dibujo rápidas y muchas impresoras láser contenían un procesador de barrido de imágenes "PostScript" (un caso especial de GPU) corriendo en un procesador RISC como el AMD 29000.

¿Cómo se comunicaban la GPU y el SO?

Como se ha podido observar a lo largo del escrito, si bien hemos abordado en gran medida a la GPU, no hemos casi mencionado al Sistema Operativo, esto se debe a una simple razón, y es que ambos no tienen una comunicación plenamente directa, ya que de forma general, el Sistema Operativo gestiona a la GPU como si fuese el periférico de la computadora. Entonces, ¿qué relación hay entre la GPU y el Sistema Operativo?, bueno resulta ser que si bien no tienen comunicación directa, e incluso desde los inicios de la GPU, esta misma hace uso de dispositivos controladores, APIs y de la CPU para poder ejecutar sus procesos plenamente. Además cabe resaltar que, como es bien sabido, al SO poseer la capacidad de gestionar los recursos de toda la computadora, también se encarga de gestionar los recursos de memoria de la propia GPU.

Así pues, y con lo anteriormente mencionado en mente, hay un cierto elemento de comunicación y funcionamiento de la GPU de antaño el cual nos parece relevante abordar nuevamente, los “framebuffer”.

Framebuffers

Un framebuffer propiamente, y como ya se ha mencionado previamente, es una porción de memoria RAM que contiene un mapa de bits que controla una pantalla de vídeo, siendo así que, se trata de un búfer de memoria que contiene datos que representan todos los píxeles de un fotograma de vídeo completo. La información del búfer suele consistir en valores de color para cada píxel que se va a mostrar en la pantalla. Los valores de color se almacenan normalmente en formato binario de 1 bit (monocromo), paletizado de 4 bits, paletizado de 8 bits, alto color de 16 bits y color verdadero de 24 bits.

El framebuffer, de igual forma y al ser parte de la memoria RAM per se, se trata de un dispositivo virtual del sistema operativo que se presenta ante las aplicaciones de diferentes maneras en función del sistema de que se hable, aunque generalmente aparece como un archivo o un bloque de memoria de acceso aleatorio reservado en la computadora, y que puede ser accedido en lectura/escritura por uno o más procesos. En este archivo o zona de memoria especial cualquier escritura modifica directamente las imágenes desplegadas en el dispositivo de vídeo, para que de esa manera los programas puedan mostrar información en pantalla sin preocuparse de los detalles de implantación, ni de la interacción real entre el ordenador y el dispositivo de vídeo.

Sin embargo, la idea de framebuffer se asocia más bien con aquellos dispositivos del sistema que son visibles o accesibles a las aplicaciones de usuario. En algunos sistemas operativos puede ocurrir que el acceso a la memoria del dispositivo de vídeo esté oculto a las aplicaciones, y que toda modificación de las imágenes desplegadas en pantalla se efectúe por medio de una API, como ocurre por ejemplo, en el caso de los sistemas operativos Windows.

La GPU y el SO

¿Cómo se comunican?

El sistema operativo, como ya se mencionó, no se comunica directamente ni con la GPU, ni con las APIs o los controladores de dispositivos (drivers) mediante "instrucciones" en el sentido tradicional de la ejecución de código de máquina. Dicho esto, un controlador de gráficos es básicamente un software diseñado

para un sistema operativo en específico que actúa como un puente entre el hardware de la GPU y el sistema operativo, permitiendo al sistema operativo interactuar con la GPU. Por ejemplo, la marca Intel desarrolla controladores de gráficos para que el sistema operativo Windows pueda comunicarse con dispositivos de gráficos Intel. De igual forma, actualizar los controladores de gráficos ayuda a resolver problemas relacionados con video o gráficos. Por ejemplo, imágenes borrosas, bajo rendimiento de gráficos en los juegos, video inestable, video entrecortado en los juegos, entre otros.

Elementos de comunicación

Finalmente, y comprendiendo que la comunicación directa entre la GPU y el SO no existe, pero que, para poder gestionar un despliegue gráfico, se utilizan interfaces de software que permiten la comunicación con estos componentes. A continuación se presenta una descripción de cómo se lleva a cabo dicha comunicación:

- Llamadas al sistema: El sistema operativo proporciona un conjunto de llamadas al sistema que permiten a las aplicaciones interactuar con los recursos del sistema, incluidos los dispositivos de hardware como la GPU. Las aplicaciones pueden realizar llamadas al sistema para acceder a funciones específicas relacionadas con la GPU, como la creación de contextos de renderizado, la carga de shaders, la transferencia de datos entre la CPU y la GPU, etc.
- Interfaz de usuario: En sistemas operativos con interfaz gráfica de usuario (GUI), como Windows o macOS, el sistema operativo proporciona una interfaz de usuario que permite a los usuarios interactuar con las aplicaciones y los recursos del sistema, incluida la GPU. Esta interfaz de usuario puede incluir herramientas de configuración y monitoreo de la GPU, como el Panel de Control de NVIDIA en Windows o las Preferencias del Sistema en macOS.
- Gestión de procesos y recursos: El sistema operativo es responsable de administrar los procesos y los recursos del sistema, incluidos los recursos de la GPU. Esto implica asignar y liberar recursos de la GPU para los procesos en ejecución, coordinar el acceso a la GPU entre múltiples

aplicaciones, y garantizar que las operaciones de la GPU se realicen de manera segura y eficiente.

Así pues, las llamadas al sistema que un sistema operativo puede hacer para interactuar con la GPU varían según el sistema operativo y las características específicas del hardware y los controladores de dispositivos. Sin embargo, algunas de las llamadas al sistema comunes relacionadas con el manejo de la GPU podrían incluir:

- Llamadas para la gestión de contextos gráficos: Permiten al sistema operativo crear, destruir y administrar contextos de renderizado en la GPU. Estos contextos proporcionan un entorno de ejecución aislado para las operaciones gráficas realizadas por las aplicaciones y los procesos.
- Llamadas para la sincronización y la programación de tareas: Permiten al sistema operativo sincronizar las operaciones de la GPU con otras operaciones en el sistema y programar tareas para su ejecución en la GPU en momentos específicos. Estas llamadas pueden incluir funciones para la espera de eventos de sincronización, la programación de colas de comandos y la gestión de la ejecución concurrente de múltiples tareas en la GPU.
- Llamadas para la gestión de errores y eventos: Permiten al sistema operativo detectar, reportar y manejar errores y eventos relacionados con la GPU y los procesos que acceden a ella. Estas llamadas pueden incluir funciones para la recuperación de errores, la notificación de eventos y la gestión de excepciones en el código de la GPU.

Referencias

- AWS. (S/F). ¿Qué es una GPU?. Recuperado de: <https://aws.amazon.com/es/what-is/gpu/>
- Alegsa, L. (2023). "Definición de Framebuffer". Recuperado de <https://www.alegsa.com.ar/Dic/framebuffer.php>
- Benj E. (08/12/08). "The computer mouse turns 40". Recuperado de <https://www.macworld.com/article/193836/mouse40.html>
- Espeso P. (14/01/14). "Las GPU como pasado, presente y futuro de la computación". Recuperado de <https://www.xataka.com/componentes/las-gpu-como-pasado-presente-y-futuro-de-la-computacion>
- Hintz E. (10/12/018). "The Mother of All Demos". Recuperado de <https://invention.si.edu/mother-all-demos>
- Facultad de Informática de Barcelona. "Historia de los videojuegos". Recuperado de <https://www.fib.upc.edu/retro-informatica/historia/videojocs.html>
- Gillis A. "DEFINITION DRAM (dynamic random access memory)". Recuperado de <https://www.techtarget.com/searchstorage/definition/DRAM>
- ICINSIGHTS (16/04/18). "Are the Major DRAM Suppliers Stunting DRAM Demand?". Recuperado de <https://web.archive.org/web/20180416202834/http://www.icinsights.com/news/bulletins/Are-The-Major-DRAM-Suppliers-Stunting-DRAM-Demand/>
- Singer, G. (7/12/2023). "The History of the Modern Graphics Processor". Recuperado de: https://www.techspot.com/article/650-history-of-the-gpu/#google_vignette
- Tibor, M. (Julio de 2023). "¿Qué es una GPU? Todo lo que necesita saber". Recuperado de: <https://softwarelab.org/es/blog/que-es-una-gpu/>
- Silberstein M. "GPUfs: The Case for Operating System Services on GPUs". Recuperado de <https://bford.info/pub/os/gpuufs-cacm.pdf>
- SunHelp. "Frame Buffer FAQ". Recuperado de <https://www.sunhelp.org/faq/FrameBuffer.html#00>
- Webopedia (24/05/21). "Frame Buffer". Recuperado de <https://www.webopedia.com/definiciones/frame-buffer/>
- Wikipedia. "History of general-purpose CPUs". Recuperado de https://en.wikipedia.org/wiki/History_of_general-purpose_CPUs