

Primer Examen - Programación de Procesadores Gráficos.

Nombre: **Horacio Iturbe García**

Sección 1: Conteste las siguientes preguntas o elija la opción correcta según corresponda.

1.- ¿Por qué son necesarias las arquitecturas multi-core (CPU) y many-core (GPU) en el computador moderno? (Valor: 5 pts)

**CPU: Se necesita un procesador de uso general, capaz de lidiar con la comunicación entre periféricos (Disco, Tarjeta de Red, teclado etc) y por supuesto con la tarjeta de gráficos**

**GPU: Se requiere de un procesador dedicado con multiples núcleos para la ejecución de tareas gráficas o bien de codificación/descodificación de audio.**

2.- ¿En lo general, un GPU puede hacer lo mismo que un CPU? Justifique. (Valor: 5 pts)

**Tomando como base genérica la ejecución de SIMD, si puede, aunque en tareas más específicas como la comunicación con la tarjeta de red ya no.**

3.- ¿En lo general, un CPU puede hacer lo mismo que un GPU? Justifique. (Valor: 5 pts)

**Tomando en cuenta la respuesta anterior, si puede aunque no tan rápido como lo haría el gpu (procesamiento de gráficos por ejemplo)**

4.- Si el rendimiento de los procesadores gráficos (GPU) es superior a los procesadores convencionales (CPU), ¿Por qué las computadoras modernas siguen necesitando a los CPU? (valor 5 pts)

**Porque se requiere de un procesador de propósito general que sea capaz de coordinar la ejecución de tareas y de asignación de recursos, además del manejo de los distintos periféricos.**

5.- Seleccione el tipo de procesador más adecuado a utilizar en las siguientes aplicaciones indicando CPU o GPU según sea el caso. Justifique su elección en cada caso. (Valor: 15 pts, 1 pt por elección correcta y justificación)

a) Procesamiento asincrónico de tareas o actividades diversificadas ( CPU)

**El procesamiento de paralelismo en el GPU es sincrónico.**

b) Control de conexiones de red en el computador (CPU )

**El CPU debe manejar además de las conexiones, los puertos de comunicación que se requieren para establecer la conexión**

c) Servidor WEB (HTTP) (CPU )

**Conexiones de red, puertos de comunicación, manejo de sistema de archivos**

d) Ejecución de un sistema operativo multitarea con prioridad ( CPU )

***Se requiere de un planificador que asigne niveles de prioridad a las tareas del sistema operativo.***

e) Ordenación de datos (CPU/GPU )

***Ambos pueden realizar ordenamiento, aunque para ordenar en un GPU, habrá que considerar el movimiento de datos del CPU al GPU***

f) Búsqueda de datos en un conjunto ordenado (CPU/GPU )

***Si ambos pueden ordenar, también pueden buscar en un conjunto ordenado***

g) Operaciones estadísticas sobre conjuntos de datos (GPU )

***Debido a la velocidad de procesamiento y de operaciones matemáticas.***

h) Virtualización de computadoras (CPU)

***Se requiere de un hypervisor que permita la asignación de recursos a la máquina virtual, además de acceso a dispositivos de red y almacenamiento***

i) Administración y operaciones sobre sistemas de archivos (CPU)

***el CPU hace uso del kernell para estas operaciones***

j) Codificación / Decodificación de video (CPU)

***Al parecer depende de si se requiere calidad o velocidad, el CPU parece ofrecer mejor calidad***

k) Procesamiento de imágenes ( GPU )

***Especializado para el procesamiento de gráficos***

l) Edición de texto ( CPU )

***Sería complicado editar texto en el GPU***

m) Codificación/Compresión de datos de audio (CPU )

***Similar a la respuesta j, el CPU parece ofrecer mejor desempeño***

n) Control de transacciones bancarias o financieras (CPU)

***Puede haber pérdida de información en el procesamiento del GPU, incluso las hay en un CPU***

o) Compresión/Descompresión sin pérdida de datos de archivos(CPU/GPU)

***Podría decir que ambos son adecuados, considero que usando ambos, el rendimiento sería mucho mejor***

No se olvide de justificar brevemente cada una de sus elecciones.

Sección II Resuelva los siguientes problemas para GPU y construya el programa para demostrarlo en cada caso usando DirectCompute Shader Model 5.0. Recolecte evidencia y anexe archivos HLSL comentado y automatización de DirectCompute del lado del CPU (fragmento código C++ donde se prepara y realiza el despacho) e imágenes o instantáneas si aplica. Sea claro y conciso.



- Calcule el centroide geométrico de la mesh de Suzanne. El archivo anexo en la actividad “Monkey.vertex” contiene la siguiente información binaria así:
  - 1 unsigned long (4byte) con el número de vértices almacenados en la secuencia.
  - N vectores (16byte) con las coordenada del vértice como un arreglo float[4]. (x,y,z,w), w=1.0f siempre.

El programa debe imprimir las coordenadas x,y,z del centroide geométrico calculado en el GPU. El CPU abre el archivo, carga los datos y los envía al GPU para su cómputo, recolecta el resultado y finalmente lo imprime en pantalla.

Dónde N: Número de vértices,  $V_i$ : cada vértice de Suzanne.

(Valor: 35 pts)

- Construya un programa que procese todo un archivo de texto grande en codificación ANSI y que convierta el contenido de minúsculas a mayúsculas. La conversión debe realizarse en GPU. El programa almacena el archivo convertido con otro nombre de archivo.

(Valor: 35 pts)

Para la entrega, suba su archivo zip, con los códigos fuente producidos para mayor comodidad junto a este documento con su nombre en el mismo zip. Por favor, utilice sólo formato zip en su entregable.