

## REPORTE DE PRÁCTICA

### IDENTIFICACIÓN DE LA PRÁCTICA

Práctica	4	Nombre de la práctica	Índices de hilos y bloques
Fecha	30/08/2021	Nombre del profesor	Alma Nayeli Rodríguez Vázquez
Nombre del estudiante		Mariana Ávalos Arce	

### OBJETIVO

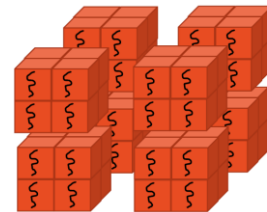
El objetivo de esta práctica consiste en implementar un kernel que muestre los índices de hilos y de bloques en una malla de tres dimensiones.

### PROCEDIMIENTO

Realiza la implementación siguiendo estas instrucciones.

Realiza un programa en C/C++ utilizando CUDA en el que implementes una malla en 3D que tenga 4 hilos en cada dimensión y 2 hilos en cada dimensión de bloque. El programa debe imprimir los valores de los siguientes índices considerando sus tres dimensiones:

- threadIdx
- blockIdx
- blockDim
- gridDim



### IMPLEMENTACIÓN

Agrega el código de tu implementación aquí.

```
#include "cuda_runtime.h"
#include "device_launch_parameters.h"
#include <stdio.h>
#include <stdlib.h>
__global__ void kernel()
{
    printf("threadIdx.x: %d, threadIdx.y: %d, threadIdx.z: %d | blockIdx.x: %d, blockIdx.y: %d,
blockIdx.z: %d | blockDim.x: %d, blockDim.y: %d, blockDim.z: %d | gridDim.x: %d, gridDim.y: %d,
gridDim.z: %d\n",
        threadIdx.x, threadIdx.y, threadIdx.z,
        blockIdx.x, blockIdx.y, blockIdx.z,
        blockDim.x, blockDim.y, blockDim.z,
        gridDim.x, gridDim.y, gridDim.z);
}
int main() {
    dim3 grid(2, 2, 2);
    dim3 block(2, 2, 2);
    kernel <<< grid, block >>> ();

    return 0;
}
```

## RESULTADOS

Agrega la imagen de la consola con el despliegue de los resultados obtenidos.

```
Microsoft Visual Studio Debug Console
threadIdx.x: 0, threadIdx.y: 0, threadIdx.z: 0 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 0, threadIdx.z: 0 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 1, threadIdx.z: 0 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 1, threadIdx.z: 0 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 0, threadIdx.z: 1 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 0, threadIdx.z: 1 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 1, threadIdx.z: 1 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 1, threadIdx.z: 1 | blockIdx.x: 0, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 0, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 0, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 1, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 1, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 0, threadIdx.z: 1 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 0, threadIdx.z: 1 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 1, threadIdx.z: 1 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 1, threadIdx.z: 1 | blockIdx.x: 1, blockIdx.y: 1, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 0, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 0, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 1, threadIdx.y: 0, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 0, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
threadIdx.x: 0, threadIdx.y: 1, threadIdx.z: 0 | blockIdx.x: 1, blockIdx.y: 0, blockIdx.z: 0 | blockDim.x: 2, blockDim.y: 2, blockDim.z: 2 | gridD
im.x: 2, gridDim.y: 2, gridDim.z: 2
```

(total de 64 líneas en consola)

## CONCLUSIONES

Escribe tus observaciones y conclusiones.

Esta práctica me facilitó bastante entender lo que en la clase pasada parecía un enredo, que fue el proceso de identificación de un hilo en una cierta configuración. Lo que más subrayaría de esta práctica es que el threadIdx sirve para localizar un hilo **dentro de su mismo bloque**, y las demás dimensiones que se imprimen confirman si la configuración de (2,2,2), tanto en malla como en bloques, sí es la correcta o no para representar la imagen. Finalmente, concluyo que las 64 líneas de salida en consola se pueden anticipar si multiplicamos 2x2x2 y luego multiplicamos ese resultado por 2x2x2, que son las dimensiones especificadas para la malla y los bloques.