

GPU PROGRAMMING

1. ¿Cuál de las siguientes configuraciones no pertenecen a la clasificación de Hwang-Briggs?

R: Computadores Multithreading.

2. El ratio entre número de operaciones en coma flotante y de accesos a la memoria global limita el rendimiento ya que:

R: Hay un ratio óptimo, valores por encima de ese ratio no mejorarían el rendimiento

3. ¿Cuál de las siguientes afirmaciones es falsa?

R: Fragment Shaders realiza las transformaciones referentes a los fragmentos tras la operación realizada por el vértice Shader.

4. ¿Cuál de las siguientes afirmaciones es falsa respecto a las Compute Capabilities?

R: La categorización incluye dos números. La primera cifra indica cambios de revisión y la segunda indica la versión.

5. Cuando se lanza un núcleo:

R: Los bloques de hilos se distribuyen de manera secuencial sobre los distintos SM.

6. ¿Cuál de las siguientes afirmaciones es falsa respecto al compilador XXXX?

R: El compilador Open64(nvopenc), que emite un lenguaje ensamblador hemos creado llamado GPU assembly.

7. Supongamos que un núcleo tiene bloques de 256 hilos, 16 instrucciones independientes de acceso a la memoria global por hilo y que cada hilo usa 8 registros y los accesos a la memoria global consumen 200 ciclos. ¿Cuántos bloques de un SM se pueden ejecutar con estas condiciones?

R: 4 bloques del SM ($8192 \text{ reg} / (256 \text{ hilos} * 8 \text{ registros/hilo})$)

8. ¿Cuál de las siguientes afirmaciones es falsa respecto a las diferentes configuraciones entre las G80, G200 y Fermi?

R: La memoria compartida en G80 y G200 es de 32 KB y en la de Fermi puede ser configurable de 32KB o de 48KB.

9. Si 6 bloques son utilizados en un SM y cada bloque tiene 128 hilos. ¿Cuántos warps hay en el SM?

R: 24

10. ¿Qué matriz es más optima?

R: 12x12 con teselado

11. ¿Cuál es el ancho de banda máximo de la tarjeta GTX 780?

R: 336 GByte/s

12. Suponiendo que tenemos un kernel que tiene 64 bloques y cada bloque tiene medio warp de hilos. ¿Cuál es la cantidad de hilos del grid? ¿Qué posición ocuparan el Hilo 12 del bloque 6?

R: 1024,108

13. En los modelos que tiene como procesador una G80. Si ejecutamos un kernel cuyos bloques de hilos consumen cada uno 7Kb de memoria compartida. ¿Cuántos bloques se podrían ejecutar a la vez?

R: Se ejecutarían 2 bloques en cada SM

14. Clipping vs culling:

R: Culling es el proceso de eliminar objetos que no son visibles por la cámara en ese marco. Clipping es el proceso por el cual los polígonos que no entran en la cámara son eliminados.

SCALA

15. Señala cuál es la salida del siguiente código:

```
def describe(x:Int) = x match {  
  case 2 => "two" case _ => "something else"  
  case true => "truth" case "hello" => "hi!"  
  case Nil => "the empty list"  
  case 5 => "five"  
  case _ => "something " }
```

si se ejecuta describe(5) Si se ejecuta describe (5) ¿qué salida proporciona?

R: Da un error de compilación

16. Las conversiones implícitas en Scala permiten:

R: Crean automáticamente objetos de una clase a partir de objetos de otra

17. La visibilidad de los elementos de una clase u objeto (métodos y atributos):

R: Funciona casi igual que en Java, tenemos los modificadores protected, private y public, y por defecto, cualquier elemento es public.

18. Señala cuál es la salida del siguiente código

```
def isEven(x: Int): Boolean =  
  if (x == 0) false else isOdd(x-1)  
def isOdd(x: Int): Boolean =  
  if (x == 0) true else isEven(x-1)
```

Ejecutar:

```
println("el valor "+isEven(4)+" el otro valor "+ !isOdd(4))
```

R: El valor false el otro valor false

19. Los parámetros de clase en Scala:

R: Son los argumentos del constructor por defecto de la clase y por lo tanto son de tipo val.

20. El tipo de la variable que proporciona el siguiente código es (ENTRADA / SALIDA):

```
def productElement(i: Int): Any = i match {
```

```
    case 0 => nombre
```

```
    case 1 => edad case _ => throw new IndexOutOfBoundsException(i.toString) }
```

R: Entrada Int / Salida String.

21. En Scala, los operadores pueden ser de tipo

R: Infijos, prefijos (mediante el empleo de sintaxis especial) y postfijo.

22. Señala cuál es la salida del siguiente código

```
val oneTwo = List(1, 2)
```

```
val threeFour = List(3, 4)
```

```
val oneTwoThreeFour = oneTwo ::: threeFour
```

```
println(oneTwo + " and " + threeFour + " were not mutated.")
```

```
println("Thus, " + oneTwoThreeFour + " is a new list.")
```

R: List(1, 2) and List(3, 4) were not mutated.

Thus, List(1, 2, 3, 4) is a new list.

23. En Scala, al escribir LIST(7,8,9):

R: Se llama al constructor de la clase List y se crea una lista con los 3 enteros indicados.

24. Señala cuál es la salida del siguiente código:

```
var x=List.range(1,10)
```

```
var evens=x.filter(_%2==0)
```

R: List(2,4,6,8).

25. En scala, la instrucción val names: Array[String] = new Array(6) define una variable de tipo Array[String], pero...

R: Los elementos de names se pueden modificar pero no se puede almacenar un nuevo array del mismo tipo en names.

26. Señala cuál es la salida del siguiente código

```
class ChecksumAccumulator {  
    val sum = 0  
}  
  
val acc = new ChecksumAccumulator  
val mul = 12  
acc.sum = 5 * mul  
println(" el acc tiene el valor de" + acc.sum )
```

R: Val no puede ser reasignada.

27. Señala cuál es la salida del siguiente programa:

```
import scala.collection.mutable.ListBuffer  
var fruits = new ListBuffer[String]()  
fruits += "Apple"  
fruits += "Banana"  
fruits += "Orange"  
fruits += ("Strawberry", "Kiwi", "Pineapple")  
fruits += "Apple"  
fruits -= ("Banana", "Orange")  
fruits -= Seq("Kiwi", "Pineapple")  
var fruitsList = fruits.toList
```

R: Ninguna de las anteriores.

R: Da error de compilación (por el Seq / Diferente del tipo esperado (String))

Si seq no está,

R1: List(Apple, Strawberry, Apple).

28. def gcd(a:Int,b:Int):Int= if (b==0) a else gcd (b,a%b)

Si ejecutamos: Print gcd(12,25)

R: El valor es 1.

29. En la API de Scala la lista vacía

R: Es un objeto de Scala llamado Nil, al añadir elementos se obtiene una lista de tipo List[A] donde A es el tipo de los elementos que se añaden.

30. Los objetos en Scala

R: Deben crearse explícitamente por el programador como cualquier otra instancia de una clase.

31. Si una clase mezcla varios traits, al ejecutarse un método definido en varios de ellos

R: Se llama al método del trait mezclado más a la derecha

```
32. def isIntIntMap(x:Any)=x match {  
    case m: Map[Int, Int]=>true  
    case _ =>false  
}
```

R: (Any) Boolean

33. Sustituye los asteriscos por su equivalente en el código:

```
package launch{ class Booster3{ }  
  
//In file bobsrockets.scala  
  
package bobsrockets{ package navigation{ package launch{ class Booster1 }  
  
class MissionControl{ val booster1= new launch.Booster1 val booster2=new  
bobsrockets.launch.Booster2 val booster3=new ***** (a rellenar) } }  
  
package launch{ class Booster2 } }
```

R: `_root_.launch.Booster3`

La otra parecida

R1: `navigator.Navigator()`

CLOUD

34. Señala cuál de las siguientes lenguajes no es soportado por Google App Engine.

R: Fortran.

35. ¿Qué tipos de despliegue permite Cloud Computing?

R: Nube privada -> destinada a un uso exclusivo por parte de la empresa

Nube pública -> pertenece a un tercer proveedor

Nube híbrida -> combinan ambas soluciones.

36. ¿Cuál de las siguientes no es/son un modelo que ofrece la Cloud Computing?:

R: IaaS, PaaS, SaaS, (On Premise).

37. ¿Indica en cuáles de las siguientes características son aportaciones de Google App Engine a los desarrolladores?

R: Limita la responsabilidad del programador

Toma el control de los picos de tráfico

Es fácilmente integrable con otros servicios de google

38. En qué modelo típico, el proveedor del servicio es responsable de la infraestructura y el administrador de BD mantiene el software y las bases de datos

R: PaaS

39. Cuando hablamos de ubiquitous computing,...

R: Integración de la informática en el entorno de la persona

R: Los ordenadores no se perciben como objetos diferenciados

40. Señala la característica falsa respecto a GWT:

R: Según el manifiesto de GWT debe ser amigable con los usuarios finales y no con los desarrolladores.

41. ¿Cuál de las siguientes afirmaciones no es correcta?:

R: Ajax escala mejor (los clientes sin estado y los servidores con estado).

42. XXXX GOOGLE WEB TOOLIKT

R: Delegar en el cliente (falta datos de la frase)

43. ¿Cuál de las siguientes afirmaciones sobre el uso de ficheros estáticos en App Engine no es correcta?

R: En java, el almacén de datos admite dos interfaces para mostrar los archivos estáticos JDO y JPA.

44. Cual de las siguientes afirmaciones sobre la planificación de tareas que se ejecutan en un determinado momento o periodo no es correcta?

R: Dentro de las posibles opciones esta realizar un backup manuales.

45. APP Engine for Java permite el uso de estándar/es para el almacen de datos:

R: Las dos anteriores

46. Señala cuál es la afirmación falsa respecto a AJAX:

R: Permite la compatibilidad entre todos los navegadores