

Tópicos Avanzados de Programación

Programación concurrente multihilo

Cuestionario Introductorio

Brambila Rivera Juan Fernando

NC: 18290858

Dar respuesta a las siguientes preguntas:

1. ¿Qué se entiende por proceso?
R: Un proceso puede entenderse informalmente como un programa en ejecución. Formalmente un proceso es "Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados".
2. ¿Qué se entiende por monoproceso?
R: Que permite un solo proceso a la vez
3. ¿Qué se entiende por multiproceso?
R: Que permite varios procesos al mismo tiempo
4. ¿Qué se entiende por ejecución paralela?
R: Se entiende como una ejecución que hace que los procesos se ejecuten uno tras otro
5. ¿Qué se entiende por ejecución concurrente?
R: Un tipo de ejecución que hace que se ejecuten 2 o mas procesos al mismo tiempo sin importar que estos se vean en pantalla o no
6. ¿Qué diferencia hay entre una ejecución paralela a una concurrente?, ¿Qué similitudes hay?
R1: La paralela hace que los procesos se ejecuten uno seguido del otro y la concurrente es que 2 o más pasen al mismo tiempo para solo generar una solución, además que, la concurrencia tiene que ver con el diseño del software, mientras que el paralelismo tiene que ver con la ejecución.
R2: Que ayudan a que la ejecución de un programa sea coordinada por el enfoque que tienen ambas y es implementada en la programación multihilos ya que permiten mediante hilos, subprocesos que se coordinan entre sí, que ocupan menos memoria que un proceso normal y comparten memoria con otros hilos.
7. ¿Qué se entiende por ejecución distribuida?
R: Un sistema distribuido es aquel en el cual sus componentes comunican y coordinan sus acciones a través de mensajes para su operación.
8. ¿Qué se entiende por subproceso?
R: Un Subproceso es un conjunto de actividades que tienen una secuencia lógica para cumplir un propósito. Un Subproceso es un Proceso por sí mismo, cuya finalidad hace parte de un Proceso más grande.
9. ¿Qué se entiende por hilo?
R: proceso ligero es una secuencia de tareas encadenadas muy pequeña que puede ser ejecutada por un sistema operativo.

10. ¿Qué se entiende por aplicaciones ejecutadas en primer plano y qué por aplicaciones ejecutadas en segundo plano?, da tres ejemplos de cada una

R: Las aplicaciones en primer plano son aplicaciones las cuales tu las tienes en pantalla utilizándolas y estas son generalmente las que usas con más concurrencia ejemplos serian: Un procesador de Texto, Una pagina de internet o un juego.

En cambio, las aplicaciones en segundo plano son aquellas que se usan sin estarlas visualizando en pantalla por decir ejecutan procesos sin que tú los veas directamente ejemplos serian: una aplicación de música, un gestor de descargas o simplemente una aplicación donde se está cargando algo.

11. ¿Qué se entiende por aplicación residente en memoria?

R: Un programa residente o TSR es un programa normal y corriente que, tras ser cargado, permanece parcial o totalmente en memoria al finalizar su ejecución. Ello es posible utilizando una función específica del sistema operativo.

12. ¿Qué se entiende por demonio dentro del contexto de una aplicación o proceso?

R: es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario. Este tipo de programas continúa en el sistema, es decir, que puede ejecutarse de forma persistente o reiniciarse si se intenta matar el proceso dependiendo de la configuración del daemon y de las políticas del sistema.

13. ¿Qué diferencia existe entre utilizar una interfaz o utilizar una clase en el momento de diseñar una aplicación?, ¿Cuándo usar una y cuando la otra?

R1: en que una interfaz lo que hace aparte de que la debes implementar esta lo que hace es que nos obliga a instanciar todos los métodos de dicha interfaz y una clase hereda para poder utilizar los atributos y los métodos.

R2: una interfaz se usa cuando necesitas utilizar los métodos y sobrescribirlos en cambio una clase la utilizas cuando necesitas sus métodos tal como ya están definidos.

14. En Java, ¿Qué clase e interface ofrece para desarrollo de aplicaciones multihilo?

R: Clase Thread. Interfaz Runnable

15. ¿Qué pasos se pudieran seguir para diseñar una aplicación basada en hilos?

R:

1. Creando un hilo:

Los s hilos se pueden crear utilizando dos mecanismos:

- Extender la clase Thread
- Implementar la interfaz Runnable

2. Ejemplo de hilo mediante la implementación de la interfaz Runnable

Aquí hay un ejemplo que crea un nuevo hilo y lo ejecuta:

```
//Crea un hilo implementando Runnable.
```

```
//Los objetos de MiHilo se pueden ejecutar en sus propios hilos
```

```
// porque MiHilo implementa Runnable.

class MiHilo implements Runnable {

    String nombreHilo;

    MiHilo(String nombre){

        nombreHilo=nombre;

    }

    //Punto de entrada del hilo

    //Los hilos comienzan a ejecutarse aquí

    public void run(){

        System.out.println("Comenzando "+nombreHilo);

        try {

            for (int contar=0; contar<10; contar++){

                Thread.sleep(400);

                System.out.println("En "+nombreHilo+", el recuento "+contar);

            }

        } catch (InterruptedException exc){

            System.out.println(nombreHilo + "Interrumpido.");

        }

        System.out.println("Terminando "+nombreHilo);

    }

}
```

```

class UsoHilos{

    public static void main(String[] args) {

        System.out.println("Hilo principal iniciando.");

        //Primero, construye un objeto MiHilo.

        MiHilo mh=new MiHilo("#1");

        //Luego, construye un hilo de ese objeto.

        Thread nuevoh=new Thread(mh);

        //Finalmente, comienza la ejecución del hilo.

        nuevoh.start();

        for (int i=0; i<50;i++){

            System.out.print(" .");

        }try{

            Thread.sleep(100);

        }catch (InterruptedException exc){

            System.out.println("Hilo principal interrumpido.");

        }

        System.out.println("Hilo principal finalizado.");

    }

}

```

3. Ejemplo de hilo al extender la clase Thread

```
class MiHilo extends Thread{

    //Construye un nuevo hilo.

    MiHilo(String nombre){

        //super se usa para llamar a la versión del constructor de Thread

        super(nombre);

    }

    //Punto de entrada del hilo

    public void run(){

        System.out.println(getName()+" iniciando.");

        //Como ExtendThread extiende de Thread, puede llamar directamente

        //a todos los métodos de Thread, incluido el método getName().

        try {

            for (int cont=0;cont<10;cont++){

                Thread.sleep(400);

                System.out.println("En "+getName()+ " , el recuento es "+cont);

            }

        }catch (InterruptedException exc){

            System.out.println(getName()+ "interrumpido.");

        }

        System.out.println(getName()+ "finalizando.");

    }

}

class ExtendThread{

    public static void main(String[] args) {

        System.out.println("Iniciando hilo principal.");

    }

}
```

```

MiHilo mh=new MiHilo("#1");

mh.start();

for (int i=0;i<50;i++){

    System.out.print(".");

    try {

        Thread.sleep(100);

    }catch (InterruptedException exc) {

        System.out.println("Hilo principal interrumpido");

    }

}

System.out.println("Hilo principal finalizado");

}

}

```

4. Crear múltiples hilos

```

class MiHilo implements Runnable{

    Thread hilo;

    //Construye un nuevo hilo.

    MiHilo(String nombre){

        hilo= new Thread(this,nombre);

    }
}

```

```
//Un método de fábrica que crea e inicia un hilo.
```

```
public static MiHilo crearYComenzar (String nombre){
```

```
    MiHilo miHilo=new MiHilo(nombre);
```

```
    miHilo.hilo.start(); //Inicia el hilo
```

```
    return miHilo;
```

```
}
```

```
//Punto de entrada de hilo.
```

```
public void run(){
```

```
    System.out.println(hilo.getName()+" iniciando.");
```

```
    try {
```

```
        for (int count=0; count<10;count++){
```

```
            Thread.sleep(400);
```

```
            System.out.println("En "+hilo.getName()+ " , el recuento es "+count);
```

```
        }
```

```
    }catch (InterruptedException exc){
```

```
        System.out.println(hilo.getName()+ " interrumpido.");
```

```
    }
```

```
    System.out.println(hilo.getName()+" terminado.");
```

```
}
```

```
}
```

```
class MasHilos {
```

```
    public static void main(String[] args) {
```

```
        System.out.println("Hilo principal iniciando.");
```

```

MiHilo miHilo1 = MiHilo.crearYComenzar("#1");

MiHilo miHilo2 = MiHilo.crearYComenzar("#2");

MiHilo miHilo3 = MiHilo.crearYComenzar("#3");


for (int i = 0; i < 50; i++) {

    System.out.print(".");

    try {

        Thread.sleep(100);

    } catch (InterruptedException exc) {

        System.out.println("Hilo principal interrumpido.");

    }

}

System.out.println("Hilo principal finalizado");

}
}

```

5. Salida de el programa

```

Hilo principal iniciando.
.#2 iniciando.
#1 iniciando.
#3 iniciando.
....En #1, el recuento es 0
En #2, el recuento es 0
...
En #3, el recuento es 9
#3 terminado.
.....Hilo principal finalizado

```


16. ¿Qué tipo de aplicaciones pudieran generarse con el uso de hilos?, mencionar mínimo 10 ejemplos (si no la saben, buscar está respuesta)

R: Aplicaciones donde se requiera el uso constante de multiprocesos por ejemplo cuando un usuario está entrando al menú otro hilo esta empezando a sacar los datos de la base de datos para darlos a un frame o a una tabla para cuando el usuario elija esa opción ya este precargado.

R2:

1. Aplicaciones de trabajo interactivo y en segundo plano
2. Procesamiento asincrónico
3. Aceleración en ejecución
4. Envío y recepción de datos
5. División de Tareas
6. Búsquedas en arboles binarios
7. Aplicaciones en tiempo real
8. Creación de relojes

17. ¿Por que es importante manejar la prioridad de los hilos (subprocesos)?

R: para que los hilos se puedan ejecutar uno después de otro básicamente para que uno de los subprocesos no se ejecute si este necesita de otro proceso para funcionar

18. ¿Qué ventajas y que desventajas podemos tener al programar con hilos?

R: Ventajas:

1. La principal ventaja de los hilos es el rendimiento. Normalmente el rendimiento es una cantidad percibida y no dada.
2. Ejecución paralela: los hilos de nivel kernel se pueden ejecutar realmente en paralelo en máquinas SMP
3. Ejecución de código asincrónico: Ya que existen múltiples hilos de ejecución en un programa construido por hilos, es posible que en otro hilo del mismo proceso sea planificado para ejecutar en el caso de que el hilo de ejecución actual se bloquee.

Desventajas:

1. Los hilos se bloquean con facilidad.
2. No todos los sistemas reconocen la existencia de hilos.
3. Requiere un mayor espacio en el núcleo para las tablas y las pilas de los hilos.

19. ¿Qué problemas se pueden generar con el uso de hilos?

R: Puede llegar a generarse un código de forma muy extensa, además de que puede que en algunos ordenadores con más núcleos de procesamiento el programa no se ejecute correctamente.

20. ¿A qué paquete pertenece la clase Thread?

R: Al paquete lang

21. Para que es utilizado el modificador de tipo `volatile`

R: Anteponiendo *volatile* a la definición de la variable `_stop` estamos diciendo al compilador que el valor contenido en esa dirección de memoria puede modificarse en cualquier momento, y queremos que dicha modificación sea visible inmediatamente para todos los hilos que están accediendo a ella, no queremos que su valor sea copiado en la caché del procesador.

Referencias:

Anónimo(21 de octubre de 2014) Usos mas comunes de los hilos.

<https://sistemasoper2.wordpress.com/2014/10/21/usos-mas-comunes-de-los-hilos/>

Pkamore V.(s.f) Ventajas de los hilos. <https://es.scribd.com/document/213528868/Ventajas-de-Los-Hilos>

Anonimo(s.f) Hilos(Threads) en java.

<http://profesores.elo.utfsm.cl/~agv/elo330/2s10/lectures/Java/threads/JavaThreads.html>

Moya Ricardo(23 de mayo de 2014) Multitarea e Hilos en Java con ejemplos (Thread & Runnable).

<https://jarroba.com/multitarea-e-hilos-en-java-con-ejemplos-thread-runnable/>

[https://es.wikipedia.org/wiki/Daemon_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Daemon_(inform%C3%A1tica))

Anonimo(s,f) Monoproceso y Multiproceso. <https://airnavas.wordpress.com/monoproceso-y-multiproceso/>