

Andy Jan

Desarrollo de aplicaciones multiplataforma

Profesor: Nuria Fuentes



Unión Europea

SEDE PORVENIR - VIAPOL: C/ Isabela, 1 - 41013 Sevilla SEDE CARTUJA: C/ Leonardo Da Vinci, 7 - 41092 Sevilla

ÍNDICE

1. ¿Qué tipo de excepción se lanza cuando el proceso falla en su ejecución?.....	1
2. Busca información sobre System.getProperty() ¿qué propiedad del sistema se usa para obtener el sistema operativo en que se ejecuta la JVM?.....	1
3. ¿Para qué nos sirve getCanonicalName() en su uso del lanzamiento de procesos Java?.....	1
4. ¿Cuántas JVM existirán en el sistema operativo si el proceso padre ejecuta sin fallos el siguiente código, siendo P1 y P2 clases del mismo paquete que ProcesoJava?.....	1
5. ¿En qué estado queda el proceso padre tras ejecutar el código anterior? ¿Puede que los subprocessos se encuentren TODOS en estado de ejecución?.....	2
6. Comprueba en la lista de procesos de sistema operativo si se lanza una nueva JVM al ejecutarse el método estático ProcesoJava.exec.....	2



```

import java.io.File;
import java.io.IOException;

public final class ProcesoJava {

    public static Process exec(Class clase) throws IOException {
        String javaHome = System.getProperty("java.home");
        String javaBin = javaHome + File.separator + "bin" + File.separator + "java";
        String classpath = System.getProperty("java.class.path");
        String className = clase.getCanonicalName();

        ProcessBuilder builder = new ProcessBuilder(javaBin, "-cp", classpath, className);
        return builder.start();
    }
}

```

1. ¿Qué tipo de excepción se lanza cuando el proceso falla en su ejecución?

El método indica “*throws IOException*” por lo tanto si ocurre algo al arrancar el proceso lanzará un IOException

2. Busca información sobre System.getProperty() ¿qué propiedad del sistema se usa para obtener el sistema operativo en que se ejecuta la JVM?

System.getProperty() devuelve una propiedad del sistema usando una Key para devolverte el valor deseado del mapa de valores.

En el caso del sistema operativo la clave es “[*os.name*](#)”

3. ¿Para qué nos sirve getCanonicalName() en su uso del lanzamiento de procesos Java?

getCanonicalName() nos sirve para obtener el nombre completo con su paquete. En el caso de lanzamiento de procesos en java es importante pasar esto para poder lanzar desde la consola para que pueda encontrar cual es la clase específica a lanzar.

4. ¿Cuántas JVM existirán en el sistema operativo si el proceso padre ejecuta sin fallos el siguiente código, siendo P1 y P2 clases del mismo paquete que ProcesoJava?

ProcesoJava.exec(P1.class); ProcesoJava.exec(P1.class); ProcesoJava.exec(P2.class);

Cada vez que lanzamos `.exec` se lanza un nuevo proceso del sistema operativo y es una nueva instancia de JVM. Por lo tanto, los tres hijos lanzados más el padre es igual a 4 JVM existiendo a la vez.

5. ¿En qué estado queda el proceso padre tras ejecutar el código anterior? ¿Puede que los subprocessos se encuentren TODOS en estado de ejecución?

Si no se usa ningún `waitFor()` todos se ejecutarán y continuarán mientras no terminen.

6. Comprueba en la lista de procesos de sistema operativo si se lanza una nueva JVM al ejecutarse el método estático `ProcesoJava.exec`.

Añadiendo un `thread.sleep()` en todas las clases incluyendo al padre para que de tiempo capturarlas en acción, aparecen las 4 ejecuciones de JVM en la lista de procesos del sistema operativo. Sin dormir el padre, este ejecuta a los 3 hijos y desaparece de la lista

```
C:\Documents and Settings\Administrador>tasklist | findstr java
java.exe          320 Console          0    11.708 KB
java.exe          1044 Console          0    11.368 KB
java.exe          2604 Console          0    11.368 KB
java.exe          2736 Console          0    11.356 KB
```