



Programación de servicios y procesos

Act2.8. Sincronización de
segmento

Francisco José García Cutillas | 2FPGS_DAM



Índice

Ejercicio 1	3
Ejercicio 2	5
Ejercicio 3	6

Ejercicio 1

Transcribe el ejemplo de la sincronización de segmentos, ejecuta el código y estudia la salida producida.

```
1 package com.mycompany.psp_act2_8;
2
3 /**
4  *
5  * @author Fran
6  */
7 public class SincronizacionSegmento extends Thread{
8
9     int id;
10    static Object bloqueo1 = new Object();
11    static Object bloqueo2 = new Object();
12
13    public SincronizacionSegmento(int id){
14
15        this.id = id;
16    }
17
18
19    public void metodol(){
20
21        synchronized (bloqueo1) {
22
23            System.out.println("Comienzo del método 1 del hilo " + id);
24
25            try{
26                Thread.sleep(1000);
27            }catch (InterruptedException ie){
28                return;
29            }
30
31            System.out.println("Fin del método 1 del hilo " + id);
32        }
33    }
34
35 }
```

```

37 public void metodo2(){
38
39     synchronized (bloqueo2) {
40
41         System.out.println("Comienzo del método 2 del hilo " + id);
42
43         try{
44             Thread.sleep(=1000);
45         }catch(InterruptedException ie){
46             return;
47         }
48
49         System.out.println("Fin del método 2 del hilo " + id);
50
51     }
52
53 }
54
55 @Override
56 public void run(){
57
58     if (id == 1) {
59
60         metodo1();
61         metodo2();
62
63     } else {
64
65         metodo2();
66         metodo1();
67
68     }
69
70 }
71
72 public static void main(String[] args){
73
74     new SincronizacionSegmento(id:1).start();
75     new SincronizacionSegmento(id:2).start();
76
77 }
78
79 }
80

```

```

-----< com.mycompany:PSP_Act2_8 >-----
Building PSP_Act2_8 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_Act2_8 ---
Comienzo del método 2 del hilo 2
Comienzo del método 1 del hilo 1
Fin del método 2 del hilo 2
Fin del método 1 del hilo 1
Comienzo del método 2 del hilo 1
Comienzo del método 1 del hilo 2
Fin del método 2 del hilo 1
Fin del método 1 del hilo 2
-----
BUILD SUCCESS
-----

Total time:  3.070 s
Finished at: 2023-11-10T08:22:13+01:00
-----

```

Ejercicio 2

Haz que los dos segmentos sincronizados utilicen el mismo objeto de bloqueo, ejecuta el código y estudia la salida producida.

```
19 public void metodo1() {
20
21     synchronized (bloqueo1) {
22
23         System.out.println("Comienzo del método 1 del hilo " + id);
24
25         try{
26             Thread.sleep(1000);
27         }catch(InterruptedException ie){
28             return;
29         }
30
31         System.out.println("Fin del método 1 del hilo " + id);
32     }
33 }
34
35
36
37 public void metodo2() {
38
39     synchronized (bloqueo1) {
40
41         System.out.println("Comienzo del método 2 del hilo " + id);
42
43         try{
44             Thread.sleep(1000);
45         }catch(InterruptedException ie){
46             return;
47         }
48
49         System.out.println("Fin del método 2 del hilo " + id);
50     }
51 }
52
53 }
```

```

-----< com.mycompany:PSP_Act2_8 >-----
Building PSP_Act2_8 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_Act2_8 ---
Comienzo del método 1 del hilo 1
Fin del método 1 del hilo 1
Comienzo del método 2 del hilo 1
Fin del método 2 del hilo 1
Comienzo del método 2 del hilo 2
Fin del método 2 del hilo 2
Comienzo del método 1 del hilo 2
Fin del método 1 del hilo 2
-----
BUILD SUCCESS
-----
Total time: 4.393 s
Finished at: 2023-11-10T08:33:36+01:00
-----

```

Ejercicio 3

Muestra las diferencias entre las dos ejecuciones y explica las razones de dichas diferencias.

En el caso del ejercicio 1, podemos observar que al utilizar objetos de bloqueo distintos, ambos métodos se sincronizan. Esto es debido a que la sincronización se realiza a nivel de método y cada método puede estar siendo ejecutado por un objeto simultáneamente. Por lo tanto se puede observar que comienzan ambos métodos al mismo tiempo y terminan al mismo tiempo.

Sin embargo, al utilizar un solo objeto de bloqueo para los dos métodos, hasta que no termine un método su ejecución, no podría empezar el siguiente.