

Programación Concurrente

UCM – Facultad de Informática
Grado en Ingeniería Informática
2012-2013
Curso 3º A

Juan Luis Álvarez Herradón
Ángel Luis Ortiz Folgado



UNIVERSIDAD COMPLUTENSE
MADRID

[MEMORIA DE LA PRÁCTICA 1]

PREGUNTAS

¿Cuáles son las ventajas y los inconvenientes de esta solución con respecto a la del primer apartado?

Banco2 es mejor que Banco1 ya que evita hacer cambios de contexto innecesarios.

En Banco1 al hacer la sincronización en la función *realizar_reintegro* y *realizar_deposito* hay que hacer el cambio de contexto y luego se puede quedar bloqueado, teniendo que hacer después otra vez un cambio de contexto para ejecutar otro hilo y luego otra vez volver a cambiar de contexto cuando el hilo se despierte.

En Banco 2 al hacer la sincronización en *hacer_reintegro* y en *hacer_deposito* en caso de que se bloquee será antes de hacer la llamada a la función y por lo tanto antes de realizar un cambio de contexto.

Si, al final de una traza de ejecución, aparece el mensaje "ERROR: discrepancia entre la liquidez calculada a partir de los depositos y reintegros (...) y el total de dinero en las cuentas (*CuentaImpl.totalSaldo()*), ¿cuál puede ser la causa? ¿Cómo se puede arreglar?

Se debe a que dos procesos pueden estar accediendo a la vez a la variable *totalSaldo*. La solución es proteger esta variable con un cerrojo para garantizar la exclusión mutua.

¿Qué efecto tendría si declaramos los métodos *hacer_reintegro* y *hacer_deposito* como métodos sincronizados?

Se garantizaría la exclusión mutua ya que no se ejecutarían hasta que no se hubiese bloqueado el cerrojo asociado a su clase.

CÓDIGO MODIFICADO Y TRAZAS

TRAZA DEL CÓDIGO ORIGINAL:

Cliente 0: deposito de 20 Euros en cuenta #0 con cajero 0
Cuenta #0: saldo es 0 Euros (pedido por Cajero 0)
Cuenta #0: actualizado a 20 Euros por cajero 0
Cliente 0: reintegro de 100 Euros en cuenta #0 con cajero 0
Cuenta #0: saldo es 20 Euros (pedido por Cajero 0)
Cuenta #0: actualizado a -80 Euros por cajero 0
ERROR: saldo negativo
Cliente 1: deposito de 50 Euros en cuenta #0 con cajero 0

APARTADO 1 - CÓDIGO:

```
class Cajero {
    public void realizar_deposito(Cuenta cta, int cantidad, Cliente
depositador) {
        synchronized(cta){
            cta.actualizarSaldo(this, cta.saldo(this) + cantidad);
            cta.notifyAll();
        }
    }

    public void realizar_reintegro(Cuenta cta, int cantidad, Cliente
reintegrador) {
        synchronized(cta){
            while((cta.saldo(this)-cantidad)<0){
                try {
                    System.out.println("El cliente "+reintegrador.id+"
está a la espera de saldo suficiente en la cuenta #" +
                                cta.numCuenta(this)+" con cajero
"+this.id);
                    cta.wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            System.out.println("El cliente "+reintegrador.id+" puede hacer
el reintegro" +
                                " en la cuenta #" +cta.numCuenta(this)+" con
cajero "+this.id);
            cta.actualizarSaldo(this, cta.saldo(this) - cantidad);
        }
    }
    ...
}
```

APARTADO 1 – TRAZA:

Cliente 1: deposito de 50 Euros en cuenta #0 con cajero 0
Cliente 2: deposito de 10 Euros en cuenta #1 con cajero 1
Cuenta #0: saldo es 0 Euros (pedido por Cajero 0)
Cuenta #1: saldo es 0 Euros (pedido por Cajero 1)
Cuenta #0: actualizado a 50 Euros por cajero 0
Cuenta #1: actualizado a 10 Euros por cajero 1
Cliente 1: deposito de 60 Euros en cuenta #0 con cajero 0
Cliente 2: reintegro de 50 Euros en cuenta #1 con cajero 1
Cuenta #0: saldo es 50 Euros (pedido por Cajero 0)
Cuenta #1: saldo es 10 Euros (pedido por Cajero 1)
El cliente 2 está a la espera de saldo suficiente en la cuenta #1 con cajero 1
Cuenta #0: actualizado a 110 Euros por cajero 0
Cliente 1: reintegro de 22 Euros en cuenta #1 con cajero 0
Cuenta #1: saldo es 10 Euros (pedido por Cajero 0)
El cliente 1 está a la espera de saldo suficiente en la cuenta #1 con cajero 0
Cliente 0: deposito de 20 Euros en cuenta #0 con cajero 0
Cliente 3: deposito de 10 Euros en cuenta #1 con cajero 1
Cuenta #0: saldo es 110 Euros (pedido por Cajero 0)
Cuenta #1: saldo es 10 Euros (pedido por Cajero 1)
Cuenta #0: actualizado a 130 Euros por cajero 0
Cuenta #1: actualizado a 20 Euros por cajero 1
Cliente 0: reintegro de 100 Euros en cuenta #0 con cajero 0
Cliente 3: reintegro de 100 Euros en cuenta #1 con cajero 1
Cuenta #0: saldo es 130 Euros (pedido por Cajero 0)
El cliente 0 puede hacer el reintegro en la cuenta #0 con cajero 0
Cuenta #1: saldo es 20 Euros (pedido por Cajero 1)
El cliente 3 está a la espera de saldo suficiente en la cuenta #1 con cajero 1
Cuenta #0: saldo es 130 Euros (pedido por Cajero 0)
Cuenta #0: actualizado a 30 Euros por cajero 0
Cliente 0: deposito de 100 Euros en cuenta #1 con cajero 0
Cuenta #1: saldo es 20 Euros (pedido por Cajero 0)
Cuenta #1: actualizado a 120 Euros por cajero 0
Cuenta #1: saldo es 120 Euros (pedido por Cajero 0)
El cliente 1 puede hacer el reintegro en la cuenta #1 con cajero 0
Cuenta #1: saldo es 120 Euros (pedido por Cajero 0)
Cuenta #1: actualizado a 98 Euros por cajero 0
Cliente 1: deposito de 200 Euros en cuenta #1 con cajero 1
Cuenta #1: saldo es 98 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 298 Euros por cajero 1
Cuenta #1: saldo es 298 Euros (pedido por Cajero 1)
El cliente 2 puede hacer el reintegro en la cuenta #1 con cajero 1
Cuenta #1: saldo es 298 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 248 Euros por cajero 1
Cliente 2: deposito de 20 Euros en cuenta #0 con cajero 1
Cuenta #1: saldo es 248 Euros (pedido por Cajero 1)
El cliente 3 puede hacer el reintegro en la cuenta #1 con cajero 1
Cuenta #0: saldo es 30 Euros (pedido por Cajero 1)
Cuenta #1: saldo es 248 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 148 Euros por cajero 1
Cuenta #0: actualizado a 50 Euros por cajero 1
Terminado con exito

APARTADO 2 – CÓDIGO:

```
class Cliente extends Thread {

    protected void hacer_deposito(int caj, int cue, int cantidad) {
        synchronized(Banco.cuentas[cue]){
            System.out.println("Cliente " + id + ": deposito de " + cantidad +
                               " Euros en cuenta #" + cue + " con cajero " + caj);

            Banco.cajeros[caj].realizar_deposito(Banco.cuentas[cue],cantidad,this);
            Banco.liquidez += cantidad;
            Banco.cuentas[cue].notifyAll();
        }
    }

    protected void hacer_reintegro(int caj, int cue, int cantidad) {
        synchronized(Banco.cuentas[cue]){
            while((Banco.cuentas[cue].saldo(Banco.cajeros[caj])-cantidad)<0){
                try {
                    System.out.println("El cliente "+this.id+" está a
la espera de saldo suficiente en la cuenta #" +

                    Banco.cuentas[cue].numCuenta(Banco.cajeros[caj])+" con cajero
"+Banco.cajeros[caj].id);
                    Banco.cuentas[cue].wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            System.out.println("Cliente " + id + ": reintegro de " + cantidad +
                               " Euros en cuenta #" + cue + " con cajero " + caj);

            Banco.cajeros[caj].realizar_reintegro(Banco.cuentas[cue],cantidad,this);
            Banco.liquidez -= cantidad;
        }
    }
}

...
}
```

APARTADO 2 – CÓDIGO:

Cliente 2: deposito de 10 Euros en cuenta #1 con cajero 1
Cliente 1: deposito de 50 Euros en cuenta #0 con cajero 0
Cuenta #1: saldo es 0 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 10 Euros por cajero 1
Cuenta #1: saldo es 10 Euros (pedido por Cajero 1)
El cliente 2 está a la espera de saldo suficiente en la cuenta #1 con cajero 1
1
Cliente 3: deposito de 10 Euros en cuenta #1 con cajero 1
Cuenta #0: saldo es 0 Euros (pedido por Cajero 0)
Cuenta #0: actualizado a 50 Euros por cajero 0
Cliente 1: deposito de 60 Euros en cuenta #0 con cajero 0
Cuenta #1: saldo es 10 Euros (pedido por Cajero 1)
Cuenta #0: saldo es 50 Euros (pedido por Cajero 0)
Cuenta #1: actualizado a 20 Euros por cajero 1
Cuenta #0: actualizado a 110 Euros por cajero 0
Cliente 0: deposito de 20 Euros en cuenta #0 con cajero 0
Cuenta #1: saldo es 20 Euros (pedido por Cajero 1)
El cliente 2 está a la espera de saldo suficiente en la cuenta #1 con cajero 1
1
Cuenta #0: saldo es 110 Euros (pedido por Cajero 0)
Cuenta #1: saldo es 20 Euros (pedido por Cajero 1)
El cliente 3 está a la espera de saldo suficiente en la cuenta #1 con cajero 1
1
Cuenta #1: saldo es 20 Euros (pedido por Cajero 0)
El cliente 1 está a la espera de saldo suficiente en la cuenta #1 con cajero 0
0
Cuenta #0: actualizado a 130 Euros por cajero 0
Cuenta #0: saldo es 130 Euros (pedido por Cajero 0)
Cliente 0: reintegro de 100 Euros en cuenta #0 con cajero 0
Cuenta #0: saldo es 130 Euros (pedido por Cajero 0)
Cuenta #0: actualizado a 30 Euros por cajero 0
Cliente 0: deposito de 100 Euros en cuenta #1 con cajero 0
Cuenta #1: saldo es 20 Euros (pedido por Cajero 0)
Cuenta #1: actualizado a 120 Euros por cajero 0
Cuenta #1: saldo es 120 Euros (pedido por Cajero 0)
Cliente 1: reintegro de 22 Euros en cuenta #1 con cajero 0
Cuenta #1: saldo es 120 Euros (pedido por Cajero 0)
Cuenta #1: actualizado a 98 Euros por cajero 0
Cliente 1: deposito de 200 Euros en cuenta #1 con cajero 1
Cuenta #1: saldo es 98 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 298 Euros por cajero 1
Cuenta #1: saldo es 298 Euros (pedido por Cajero 1)
Cliente 3: reintegro de 100 Euros en cuenta #1 con cajero 1
Cuenta #1: saldo es 298 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 198 Euros por cajero 1
Cuenta #1: saldo es 198 Euros (pedido por Cajero 1)
Cliente 2: reintegro de 50 Euros en cuenta #1 con cajero 1
Cuenta #1: saldo es 198 Euros (pedido por Cajero 1)
Cuenta #1: actualizado a 148 Euros por cajero 1
Cliente 2: deposito de 20 Euros en cuenta #0 con cajero 1
Cuenta #0: saldo es 30 Euros (pedido por Cajero 1)
Cuenta #0: actualizado a 50 Euros por cajero 1
Terminado con exito