Proceso 1: Escritor (FileWriterProcess)

Este proceso creará un archivo de bloqueo (lockfile.lock) para indicar que está escribiendo en un archivo de datos (data.txt). Al finalizar la escritura, eliminará el archivo de bloqueo.

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class FileWriterProcess {

  private static final String LOCK_FILE = "lockfile.lock";
  private static final String DATA_FILE = "data.txt";

  public static void main(String[] args) {
    File lockFile = new File(LOCK_FILE);

    try {
      // Crear el archivo de bloqueo
      if (lockFile.createNewFile()) {
        System.out.println("Archivo de bloqueo creado. Comenzando la escritura de datos...");

        // Crear el archivo de datos y escribir información en él
        FileWriter writer = new FileWriter(DATA_FILE);
        writer.write("Datos importantes generados por el proceso de escritura.\n");
        writer.write("Línea 1: Información relevante.\n");
        writer.write("Línea 2: Más datos interesantes.\n");
        writer.close();

        System.out.println("Escritura completada. Eliminando el archivo de bloqueo...");
        lockFile.delete();  // Eliminar el archivo de bloqueo al finalizar la escritura
        System.out.println("Archivo de bloqueo eliminado. Proceso de escritura finalizado.");
      } else {
        System.out.println("El proceso de escritura ya está en ejecución o el archivo de bloqueo no pudo ser
creado.");
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

Proceso 2: Lector (FileReaderProcess)

Este proceso esperará a que el archivo de bloqueo lockfile.lock desaparezca, lo que indica que el proceso de escritura ha terminado. Luego leerá el contenido del archivo de datos data.txt.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

public class FileReaderProcess {

  private static final String LOCK_FILE = "lockfile.lock";
  private static final String DATA_FILE = "data.txt";

  public static void main(String[] args) {
    File lockFile = new File(LOCK_FILE);
```

```
        // Esperar a que el archivo de bloqueo desaparezca
        System.out.println("Esperando a que se libere el archivo de bloqueo...");
        while (lockFile.exists()) {
            try {
                Thread.sleep(1000); // Pausa por 1 segundo antes de verificar de nuevo
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Leer el archivo de datos
        System.out.println("Archivo de bloqueo liberado. Leyendo el archivo de datos...");

        try (BufferedReader reader = new BufferedReader(new FileReader(DATA_FILE))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println("Leído: " + line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        System.out.println("Proceso de lectura completado.");
    }
}
```

Explicación

1. Proceso de Escritura (FileWriterProcess):
    - Crea un archivo de bloqueo lockfile.lock para indicar que está trabajando.
    - Escribe en un archivo de datos data.txt.
    - Al finalizar la escritura, elimina el archivo de bloqueo lockfile.lock para señalar que ha terminado.
2. Proceso de Lectura (FileReaderProcess):
    - Espera a que el archivo de bloqueo lockfile.lock desaparezca.
    - Lee el archivo de datos data.txt y muestra su contenido en la consola.

Instrucciones para Ejecutar

Para simular la sincronización, puedes ejecutar ambos procesos de las siguientes maneras:

1. **Desde la línea de comandos**:
    - Abre dos terminales.
    - En una terminal, ejecuta el **proceso de escritura**.
    - En la otra terminal, ejecuta el **proceso de lectura**.

Así podrás ver cómo el proceso de lectura espera hasta que el proceso de escritura termine antes de leer el archivo.

2. **Mediante un script batch o shell**: Puedes crear un script que ejecute ambos procesos simultáneamente o de manera controlada.

Cómo Ejecutar Simultáneamente los Procesos

Vamos a mostrar cómo hacer esto en un script para sistemas operativos basados en Unix (como Linux o macOS) usando un script .sh o en Windows con un script .bat.

Script en Unix (Linux/macOS)

1. Crea un archivo llamado run_processes.sh con el siguiente contenido:

```
#!/bin/bash
# Compilar los programas Java
javac FileWriterProcess.java
javac FileReaderProcess.java
# Ejecutar el proceso de escritura en segundo plano (&)
java FileWriterProcess create &

# Ejecutar el proceso de lectura simultáneamente
java FileReaderProcess
```

2. Asigna permisos de ejecución al script y ejecútalo:

```
chmod +x run_processes.sh
./run_processes.sh
```

Script en Windows (.bat)

1. Crea un archivo llamado run_processes.bat con el siguiente contenido:

```
@echo off
rem Compilar los programas Java
javac FileWriterProcess.java
javac FileReaderProcess.java

rem Ejecutar el proceso de escritura en segundo plano
start java FileWriterProcess create

rem Ejecutar el proceso de lectura en otra ventana de consola
start java FileReaderProcess
```

2. Ejecuta el script run_processes.bat en la línea de comandos de Windows.

Ejecución Simultánea Manual

Si prefieres ejecutarlo manualmente en dos terminales diferentes:

1. **Primera Terminal**:

- Ejecuta el proceso de escritura:

```
java FileWriterProcess create
```

2. **Segunda Terminal**:
   - Ejecuta el proceso de lectura:

```
java FileReaderProcess
```

Cómo Funciona?

- El **proceso de escritura** se ejecutará primero, creando el archivo de bloqueo (lockfile.lock) y escribiendo en data.txt.
- El **proceso de lectura** esperará mientras el archivo de bloqueo esté presente.
- Una vez que el proceso de escritura termine (elimina lockfile.lock), el proceso de lectura detectará que el archivo de bloqueo ya no existe y procederá a leer el archivo data.txt.