Ejecución de programas con Java ProcessBuilder

Se ejecutarán programas del SO con el método “command”. El método “waitFor” se utiliza para esperar a que el proceso finalice.

```
import java.io.IOException;

public class ExecuteProgram {

    public static void main(String[] args) throws IOException,
InterruptedException {

        var processBuilder = new ProcessBuilder();

        processBuilder.command("notepad.exe");

        var process = processBuilder.start();

        var ret = process.waitFor();

        System.out.printf("Program exited with code: %d", ret);
    }
}
```

Este programa lanza el Bloc de Notas de Windows. Retornará un código al finalizar su ejecución.

Sincronización de procesos con Java ProcessBuilder

Lanzamos dos programas con el método “command”. Con la ayuda de “waitFor” esperamos a que ambos programas/procesos se ejecuten.

```
public class TwoReaders {

    public static void main(String[] args) throws Exception {

        System.out.println("Two processes start synchronously");

        ProcessBuilder pa = new ProcessBuilder("notepad.exe",
"C:\\book1.txt");
        ProcessBuilder pb = new ProcessBuilder("notepad.exe",
"C:\\book2.txt");
        Process p1 = pa.start();
        Process p2 = pb.start();

        p1.waitFor();
        p2.waitFor();

        System.out.println("Wait until both processes completed!");
    }
}
```

Java ProcessBuilder: Salida de datos al ejecutar comandos.

Este ejemplo ejecuta un comando y muestra su salida.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class ProcessBuilderEx {

    public static void main(String[] args) throws IOException {

        var processBuilder = new ProcessBuilder();

        processBuilder.command("cal", "2019", "-m 2");

        var process = processBuilder.start();

        try (var reader = new BufferedReader(
            new InputStreamReader(process.getInputStream()))) {

            String line;

            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }

        }
    }
}
```

Ejemplo para lanzar el comando “cal” en Linux.

```
processBuilder.command("cal", "2019", "-m 2");
```

Este comando ejecuta el programa “cal”, los otros parámetros son las opciones del programa. Para lanzar un comando en un sistema Windows, podermos utilizar lo siguiente: processBuilder.command("cmd.exe", "/c", "ping -n 3 google.com").

```
var process = processBuilder.start();
```

El proceso es lanzado con el método “start”.

```
try (var reader = new BufferedReader(
    new InputStreamReader(process.getInputStream()))) {
```

Mediante el método “getInputStream” podemos acceder al flujo de entrada desde la salida estándar del proceso.

Java ProcessBuilder redirección de flujo de salida de datos

Con redirectOutput, redirigimos el flujo de salida por defecto de la clase “processBuilder”.

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

public class RedirectOutputEx {

    public static void main(String[] args) throws IOException {

        var homeDir = System.getProperty("user.home");

        var processBuilder = new ProcessBuilder();

        processBuilder.command("cmd.exe", "/c", "date /t");

        var fileName = new File(String.format("%s/Documents/tmp/output.txt",
homeDir));

        processBuilder.redirectOutput(fileName);

        var process = processBuilder.start();

        try (var reader = new BufferedReader(
                new InputStreamReader(process.getInputStream()))) {

            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        }
    }
}
```

El programa redirige la salida a un fichero. Lanza el comando de fecha de Windows.

```
processBuilder.redirectOutput(fileName);
```

Ahora redirigimos la salida estándar de processBuilder a un fichero:

```
try (var reader = new BufferedReader(
    new InputStreamReader(process.getInputStream()))) {

    String line;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
}
```

Observamos que la salida ha ido a parar a un fichero.

```
$ echo %cd%
C:\Users\Jano\Documents\tmp
$ more output.txt
Thu 02/14/2019
```

Se ha escrito la fecha en el fichero output.txt file.

Java ProcessBuilder para redirigir la entrada y salida de datos

El siguiente ejemplo redirige salida y entrada de datos. Partimos de los datos del fichero “input.txt”

```
sky
blue
steel
morning
coffee
earth
forest
```

Este es el contenido del fichero

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

public class ProcessBuilderRedirectIOEx {

    public static void main(String[] args) throws IOException {

        var processBuilder = new ProcessBuilder();

        processBuilder.command("cat")
                .redirectInput(new File("src/resources", "input.txt"))
                .redirectOutput(new File("src/resources/", "output.txt"))
                .start();
    }
}
```

En este programa, se redirige la entrada de datos desde un fichero input.txt a través del comando y la salida de datos va a parar a un fichero “outputs.txt”

Java ProcessBuilder para cambiar directorio de trabajo

El método “directory” establece el directorio de trabajo para “processBuilder”.

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

public class ProcessBuilderDirectoryEx {

    public static void main(String[] args) throws IOException {

        var homeDir = System.getProperty("user.home");

        var pb = new ProcessBuilder();

        pb.command("cmd.exe", "/c", "dir");
        pb.directory(new File(homeDir));

        var process = pb.start();

        try (var reader = new BufferedReader(
                new InputStreamReader(process.getInputStream()))) {

            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        }
    }
}
```

El ejemplo anterior establece el directorio raíz como directorio de trabajo de processBuilder. Después se muestra el contenido del directorio raíz en pantalla.

```
var homeDir = System.getProperty("user.home");
```

Con esto comprobamos el directorio raíz del usuario

```
pb.command("cmd.exe", "/c", "dir");
```

Así hemos definido un proceso que ejecuta el comando en Windows. Ahora establecemos el directorio de trabajo para “processBuilder”

```
pb.directory(new File(homeDir));
```

Obtendremos en pantalla:

```
Volume in drive C is Windows
Volume Serial Number is 4415-13BB

Directory of C:\Users\Jano

02/14/2019  11:48 AM    <DIR>          .
02/14/2019  11:48 AM    <DIR>          ..
10/13/2018  08:38 AM    <DIR>          .android
01/31/2019  10:58 PM               281 .bash_history
12/17/2018  03:02 PM    <DIR>          .config
...
```