

Gestión de procesos y redirección de salida en Kotlin

Lectura de la salida de un proceso con `BufferedReader` e `InputStreamReader`

[crayon-6a9d57d42e034376983933/]

Explicación:

- En este ejemplo, utilizamos `ProcessBuilder` para ejecutar un comando en el sistema operativo y luego capturamos y mostramos su salida.
- `InputStreamReader` se utiliza para convertir la salida del proceso en una secuencia de caracteres legibles.
- `BufferedReader` se utiliza para leer la salida del proceso línea por línea.
- El bucle `while` lee y muestra cada línea de salida hasta que llega al final de la salida.
- `proceso.waitFor()` se utiliza para esperar a que el proceso termine y devuelve el estado de salida del proceso (0 si se ejecutó correctamente).

Teoría:

- `ProcessBuilder` se utiliza para crear y controlar procesos en el sistema operativo desde una aplicación Java/Kotlin.
- `InputStreamReader` se utiliza para convertir una secuencia de entrada de bytes (como la salida de un

proceso) en una secuencia de entrada de caracteres que se puede leer con facilidad.

- `BufferedReader` se utiliza para leer texto de una secuencia de entrada de caracteres de manera eficiente, ya que almacena en búfer caracteres y realiza lecturas más eficientes en términos de rendimiento en comparación con la lectura de caracteres uno por uno.
-

Redirección de la salida de un proceso a un archivo

[crayon-6a9d57d42e03a892821246/]

Explicación:

- En este ejemplo, utilizamos `ProcessBuilder` para ejecutar un comando y redirigir su salida estándar a un archivo específico.
- `ProcessBuilder.Redirect.to(File(outputFile))` se utiliza para especificar que la salida del proceso debe ser redirigida al archivo cuyo nombre se encuentra en la variable `outputFile`.
- Después de ejecutar el proceso, esperamos a que termine y verificamos su estado de salida.

Teoría:

- La redirección de la salida de un proceso a un archivo es útil para capturar la salida del proceso y guardarla en un archivo en lugar de mostrarla en la consola.
- `ProcessBuilder.Redirect.to(File(outputFile))` es una forma de redirigir la salida estándar de un proceso en Java/Kotlin. La salida se escribirá en el archivo

especificado en outputFile.

Heredar la salida de un proceso a la consola

[crayon-6a9d57d42e03d627659586/]

Explicación:

- En este ejemplo, utilizamos `ProcessBuilder` para ejecutar un comando y heredar su salida estándar, lo que significa que la salida se mostrará en la consola.
- `ProcessBuilder.Redirect.INHERIT` se utiliza para especificar que la salida del proceso debe heredarse y mostrarse en la consola.

Teoría:

- La herencia de la salida de un proceso a la consola es útil cuando deseas que la salida del proceso se muestre directamente en la consola en lugar de redirigirla a un archivo o capturarla en tu programa.
- `ProcessBuilder.Redirect.INHERIT` es una opción que permite heredar la salida estándar del proceso y mostrarla en la consola en lugar de redirigirla o capturarla en el programa.