Realizar una comunicación entre dos procesos en Kotlin (un proceso envía información a otro proceso y el otro proceso responde)

Padre (envía información al proceso hijo y recibe la respuesta del hijo)

[crayon-6a9d61a6496e2910547588/]

Este código en Kotlin crea un proceso hijo utilizando la clase ProcessBuilder, envía un mensaje al proceso hijo a través de su flujo de salida estándar y lee la respuesta del proceso hijo desde su flujo de entrada estándar. Aquí está la explicación línea por línea:

1. **Importaciones de clases necesarias**: Se importan las clases BufferedReader, InputStreamReader, OutputStreamWriter para manejar las operaciones de entrada y salida del proceso hijo.
2. **Función principal main**: Se define la función main, que es el punto de entrada del programa.
3. **Creación del ProcessBuilder**: Se crea un objeto ProcessBuilder con los parámetros necesarios para ejecutar el proceso hijo. En este caso, el comando a ejecutar es java -cp /ruta/a/la/clase/principal/del/proceso/hijo HijoKt, donde /ruta/a/la/clase/principal/del/proceso/hijo es la

ruta a la clase principal del proceso hijo y HijoKt es el nombre de la clase principal del proceso hijo.
4. **Inicio del proceso hijo**: Se inicia el proceso hijo utilizando el ProcessBuilder.
5. **Escritura en el flujo de salida del proceso hijo**: Se obtiene un OutputStreamWriter a partir del flujo de salida (outputStream) del proceso hijo. Se escribe un mensaje en el flujo de salida del proceso hijo utilizando el método write, seguido de un salto de línea (\n). Se llama al método flush para asegurarse de que todos los datos se envíen al proceso hijo.
6. **Lectura de la salida del proceso hijo**: Se obtiene un BufferedReader a partir del flujo de entrada (inputStream) del proceso hijo. Se lee una línea de la salida del proceso hijo utilizando el método readLine y se almacena en la variable outputMessage.
7. **Impresión del mensaje recibido**: Se imprime el mensaje recibido del proceso hijo en la consola.

Hijo (el hijo recibe el mensaje del padre y responde al padre)

[crayon-6a9d61a6496ea128118144/]

Este código en Kotlin lee una entrada del usuario desde la consola, agrega un mensaje adicional y luego escribe el resultado en la salida estándar. Aquí está la explicación línea por línea:

1. **Importaciones de clases necesarias**: Se importan las clases BufferedReader, InputStreamReader, y OutputStreamWriter para manejar las operaciones de entrada y salida.
2. **Función principal main**: Se define la función main, que es el punto de entrada del programa.

3. **Creación del lector de entrada**: Se crea un objeto BufferedReader para leer la entrada del usuario desde la consola (System.in). Esto permite al usuario ingresar datos desde la consola.
4. **Creación del escritor de salida**: Se crea un objeto OutputStreamWriter para escribir en la salida estándar (System.out). Esto permite escribir en la consola.
5. **Lectura de la entrada del usuario**: Se lee una línea de entrada del usuario utilizando el método readLine() del BufferedReader y se almacena en la variable parentMessage.
6. **Escritura en la salida estándar**: Se escribe en la salida estándar una concatenación del mensaje del usuario y la cadena «SOY HIJO», utilizando el método write() del OutputStreamWriter. Se llama al método flush() para asegurarse de que todos los datos se envíen a la salida estándar.

En resumen, este código espera que el usuario ingrese una línea de texto desde la consola, luego agrega la cadena «SOY HIJO» al final de esa línea y la imprime de vuelta en la consola. Esto ilustra cómo interactuar con la entrada y salida estándar en un programa Kotlin.