

Comunicar dos procesos en Kotlin

Si deseas comunicar dos procesos en el mismo equipo sin utilizar sockets, una opción es utilizar la comunicación interprocesos (IPC, por sus siglas en inglés) a través de pipes o tuberías. En Kotlin, puedes utilizar la clase `Process` para crear y controlar procesos.

En este ejemplo, el proceso padre crea un proceso hijo utilizando `ProcessBuilder` y establece la salida estándar del proceso hijo como la entrada del proceso padre. Luego, lee la salida del proceso hijo y muestra el mensaje recibido.

```
[crayon-6a9d58015f780662648440/]
```

Por otro lado, necesitarás crear un archivo separado llamado «`ChildProcess.kt`» que contenga el código del proceso hijo. Aquí tienes un ejemplo básico de cómo podría ser el código del proceso hijo:

```
[crayon-6a9d58015f786050508534/]
```

En el código del proceso hijo, se escribe el mensaje en la salida estándar utilizando `OutputStreamWriter`.

Recuerda reemplazar «`ChildProcess`» por el nombre del archivo de la clase del proceso hijo que creaste.

Este es solo un ejemplo básico de cómo comunicar dos procesos en el mismo equipo sin utilizar sockets en Kotlin. Ten en cuenta que existen otras técnicas de IPC disponibles, como archivos compartidos, memoria compartida, colas de mensajes, entre otras, que podrían adaptarse mejor a tus necesidades específicas.

