

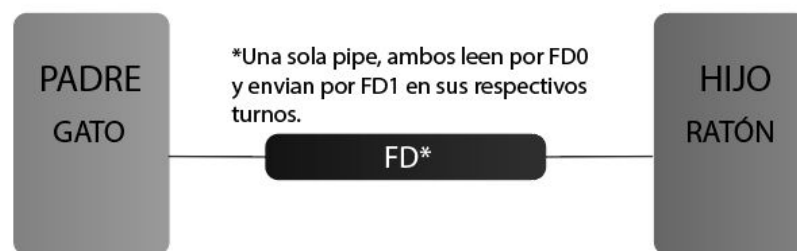
Se han realizado todos los apartados de la práctica, incluso el punto de excelencia.

Memoria:

1. Indica qué procesos se utilizan en tu práctica y cuál es la jerarquía.

En la práctica hemos utilizado dos procesos. El primer proceso tiene la función de servidor y jugador y por lo tanto actúa como padre y gato. El segundo proceso únicamente es jugador y actúa como ratón.

2. Indica cómo queda la tabla de file descriptors de cada uno de los procesos y para qué se utiliza cada uno.



3. Explica en qué momentos se envía la información y dónde se recibe. Relaciónalo con la zona del código dónde ocurra (indicar la función o la línea de código).

Después de realizar el movimiento validado de las piezas se envía la información y mediante un signal se informa al otro proceso de leer la pipe.

En el caso del padre/gato se escribe la pipe en la línea 313 y se informa al hijo en la 314. En el caso del hijo/ratón se escribe la pipe en la línea 240 y se informa al padre en la 241.

Ambos leen la pipe cuando reciben un signal. Para el padre/gato es la función PadreLeeHijo de la línea 87 y para el hijo/ratón es la función HijoLeePadre de la línea 105.

4. Explica si has utilizado signals y para qué.

Si, hemos utilizado los dos signals del usuario. Cada uno de ellos para indicarle al otro proceso que debe de leer las pipes.

5. Explica si has utilizado threading y para qué.

No hemos creído conveniente usar threads dado que el juego se ejecuta por turnos.

6. Conclusiones: Qué has aprendido con esta práctica, qué te ha costado más, cualquier opinión que consideres relevante...

Esta práctica nos ha ayudado a consolidar los fundamentos de sistemas operativos que hemos aprendido estas primeras semanas en clase. Gracias a que disponíamos de gran parte del código y que estaba comentado no nos ha resultado complicado trabajar con la librería SFML, tampoco hemos encontrado dificultades en la comunicación entre procesos ni en idear una lógica para almacenar la posición de la fichas en el tablero. La mayor dificultad ha sido, o creemos que ha sido, por errores del programa/lenguaje en sí, dado que intentamos probar el funcionamiento del read y write en el main y no nos funcionaba (luego al copiarlo en la función de dibuja, sin modificar nada, si funcionaba), lo mismo nos pasaba al intentar debugar el código, ya que si no escribíamos `std::endl` al final de cada `cout` no imprimía nada por pantalla, lo que comportó una gran pérdida de tiempo intentando descubrir porque no entraba en ciertas funciones cuando realmente si lo hacía.