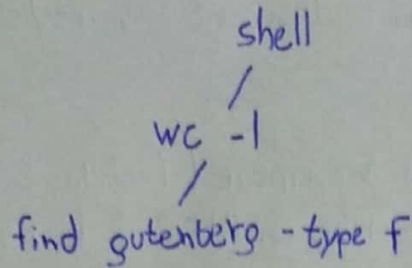


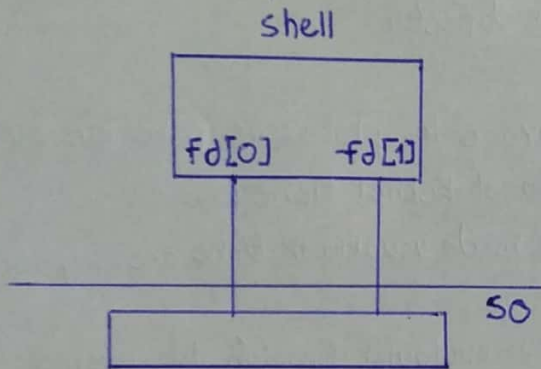
Problema 1)

1. La estructura de padre-hijos que se genera es la siguiente:

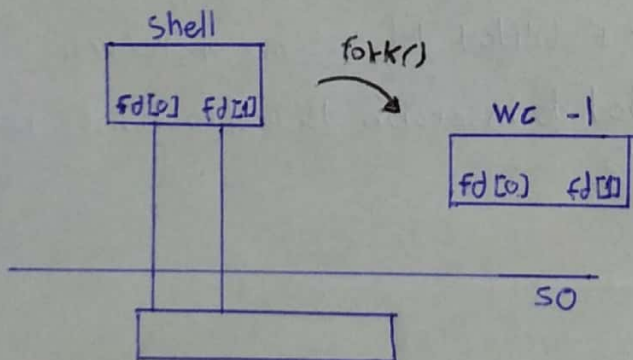


2. Las dos partes del código se ejecutan a la vez, ya que el padre (`wc -l`) no hace ningún "wait" para esperar que el proceso hijo (`find`) se acabe. Al final del código se hace un "wait" para esperar que el proceso `wc -l` (hijo del `shell`) se acabe.

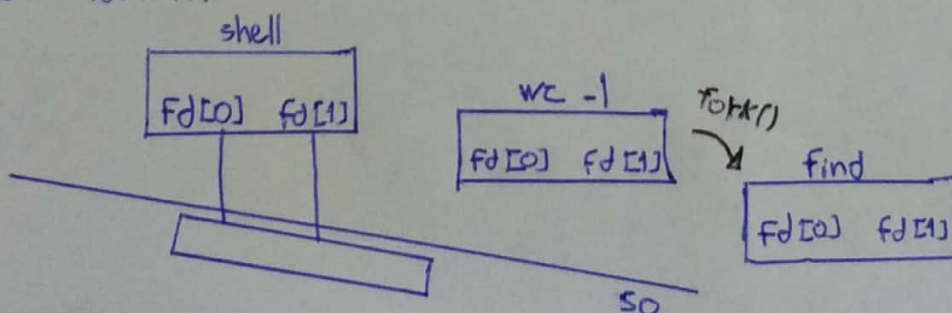
3. Se crea una tubería, que es un "buffer" interno del sistema operativo. La tubería hace que `fd[1]` permita escribir en el "buffer" y `fd[0]` permita leer del "buffer".



4. Se crea un nuevo proceso, que es una copia del padre "shell", mediante la llamada a la función `fork()`.



5. Se crea un nuevo proceso, que es una copia del padre "wc -l", mediante la llamada a la función `fork()`.



6. En este punto hay las comunicaciones padre-hijo: "shell" - "wc -l" y "wc -l" - "find". Así, los procesos pueden comunicarse por señales, como por ejemplo esperar que el proceso hijo se ocabe para que el proceso padre continúe mediante el uso de las funciones `wait(NULL)` y `exit()`.
7. A la línea 12 del código se asocia el `fd[1]` al descriptor 1 (salida estándar). El objetivo es redireccionar todo lo que se escribe en `fd[1]` a la pantalla del terminal.
8. A la línea 19 del código se asocia el `fd[0]` al descriptor 0 (entrada estándar). El objetivo es conseguir leer todo lo que se ha escrito el comando 'find' en el terminal.
9. Lo que hace el código es contar cuántos ficheros hay en la carpeta "gutenberg". El comando equivalente en el bash es: `Find gutenberg -type f | wc -l`
10. El comando "wc -l" lo imprime porque la respuesta de un proceso está asociada a la salida estándar (pantalla del terminal) por defecto.

Problema 2)

1. Es cualquier función que ofrece el sistema operativo a los procesos para que puedan acceder a los dispositivos o hacer operaciones que solo el Kernel tiene acceso. Son necesarias para realizar las operaciones que el modo usuario no tiene acceso, como por ejemplo abrir un archivo.
2. Las consecuencias serían la imposibilidad de ejecutar cualquier función del sistema operativo ya que no tendríamos el nivel de acceso necesario.
3. Es una situación inesperada que ocurre en la ejecución de un proceso. Son necesarias para la seguridad y fiabilidad del sistema operativo.
4. Las consecuencias serían la posibilidad de inserción de instrucciones peligrosas, pérdida de datos, entre otros problemas.